A Novel Battery Management & Charging Solution

for Autonomous UAV Systems

by

Sami Mian

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2018 by the
Graduate Supervisory Committee:

Sethuraman Panchanathan, Chair
Spring Berman
Yezhou Yang
Troy McDaniel

ARIZONA STATE UNIVERSITY

May 2018

ABSTRACT

Currently, one of the biggest limiting factors for long-term deployment of autonomous systems is the power constraints of a platform. In particular, for aerial robots such as unmanned aerial vehicles (UAVs), the energy resource is the main driver of mission planning and operation definitions, as everything revolved around flight time. The focus of this work is to develop a new method of energy storage and charging for autonomous UAV systems, for use during long term deployments in a constrained environment. We developed a charging solution that allows pre-equipped UAV system to land on top of designated charging pads and rapidly replenish their battery reserves, using a contact charging point. This system is designed to work with all types of rechargeable batteries, focusing on Lithium Polymer (LiPo) packs, that incorporate a battery management system for increased reliability. The project also explores optimization methods for fleets of UAV systems, to increase charging efficiency and extend battery lifespans. Each component of this project was first designed and tested in computer simulation. Following positive feedback and results, prototypes for each part of this system were developed and rigorously tested. Results show that the contact charging method is able to charge LiPo batteries at a 1-C rate, which is the industry standard rate, maintaining the same safety and efficiency standards as modern day direct connection chargers. Control software for these base stations was also created, to be integrated with a fleet management system, and optimizes UAV charge levels and distribution to extend LiPo battery lifetimes while still meeting expected mission demand. Each component of this project (hardware/software) was designed for manufacturing and implementation using industry standard tools, making it ideal for large-scale implementations. This system has been successfully tested with a fleet

of UAV systems at Arizona State University, and is currently being integrated into an Arizona smart city environment for deployment.

ACKNOWLEDGMENTS

I would first like to express my sincere gratitude to my thesis advisor and committee chair, Dr. Sethuraman Panchanathan of the School of Computing, Informatics, and Decisions Systems Engineering at Arizona State University, for his continuous support of my Master's study and research, by allowing me to join his research group to complete my research, and by funding my project and education. Dr. Panch has provided a significant amount of independence with this project, allowing me to define and implement my original ideas and supporting my decisions along the way.

I would next like to thank Dr. Spring Berman of the School for Engineering of Matter, Transport, and Energy, for serving as the second reader of my thesis. She has allowed me to work alongside other students within her research group, where I have learned the fundamentals of control theory and long-term robot deployment strategies.

I would also like to acknowledge Dr. Yezhou Yang the School of Computing, Informatics, and Decisions Systems Engineering at Arizona State University as the third reader of this thesis, and I am gratefully indebted to him for his very valuable comments on this thesis.

I would also like to acknowledge Dr. Troy McDaniel of the School of Computing, Informatics, and Decisions Systems Engineering as the fourth member of my committee for this thesis. His door has always been open for advice and guidance, and his help with content and organization reviews has been invaluable in completing this paper.

Next I would like to thank Mark Naufel of the Office of Knowledge Enterprise and Development, for his mentorship, support, guidance, and friendship. He has helped enable

this project at a high level, providing physical and monetary support, and has given me a platform to implement all of these ideas without obstacle.

I would like to extend a huge thanks to John Patterson, an undergraduate electrical engineering student who has been aiding me on this project since November of 2017. John has been instrumental in the development of the power management system, helping choose industrial-grade components and aiding all of the testing runs for data collection.

Lastly, I would like to thank the members of the ASU Luminosity Lab, in particular Glenn Pace and Daniel D'Souza, who have helped make this project a reality. They have been critical in helping develop the original designs for the prototype, as well as the first build for testing.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Automated systems are becoming more common in everyday life as the field of robotics and automation continues to advance. From the introduction of the robot vacuum cleaner for the home environment, to ongoing development of self-driving technology in the automotive world, new advances in robotics technologies are being used to improve efficiency in both a business and consumer setting. However, there are several large barriers that prevent robots from taking their next big leap into widespread utilization. One of the big challenges is using robotic platforms for extended periods of time, usually in diverse environments where resources can be scarce (Bellingham, 2007). A perfect example is using unmanned aerial vehicles (UAVs) for surveillance in remote area, such as a battlefield. Another is the use of robots in a smart city environment, being constantly deployed for different tasks such as security, safety, and public service. In order to make these use cases a reality, there are a few roadblocks that need to be solved. One of the biggest current problems with robots is their operating limitations due to power constraints (Goldstein, 2004). Most robots, whether they are simple vacuum cleaners or highly advanced military drones, are constrained to the limits of their power source; when their battery or fuel source depletes, they need to be recharged or refueled. Replacing a battery or refilling a fuel tank takes time and resources, and is not always easily accomplishable. For some long-term missions, it could take weeks before personnel can reach the platform in question to perform these tasks. And for robots that are deployed in areas that are hostile environments or inaccessible, sending in a person to provide this maintenance is next to impossible. In order to allow for future possibilities for long term use of robots in various

environments, both friendly and hostile, this major problem with power systems needs to be solved (Campbell 2005). A novel type of power system, both storage and recharging, needs to be developed, since the key to efficiency in these cases is battery/power longevity.

There are several ways in which power problems are traditionally solved with robots. For many, the solution is to put a bigger battery on the robot or find a more efficient power source. But in some cases, such as airborne robots or UAVs, weight and size is a critical factor that affects performance. For a UAV system, the more weight added to a system (such as a bigger battery) the more power is required for the vehicle to stay inflight. Very quickly, you hit a point of diminishing returns, where the increased mass of the vehicle outweighs the power benefits. So, the question is how to optimize both battery size and lifetime in order to get the most efficient use of the system.

One of the most interesting environments where this can be a huge advancement is the smart city environment. Arizona State University is the perfect location for this type of technology, in part because the university is currently laying the groundwork to turn the Tempe campus into a smart city. A majority of the focus right now is to update existing technologies around campus, such as converting all emergency boxes to VoIP systems, but there are also some efforts to develop new technologies for novel solutions. One of the main university-wide initiatives is the integration of a fleet of autonomous UAV systems into the campus security system. The goal of this project, referred to as Project Airbud, is to help the ASU Police Department create a safer environment for the ASU community.

The focus of this research has been to develop a unique, modern, efficient battery charging solution and maintenance for a fleet of autonomous unmanned aerial vehicles (UAVs). The main testing site is on the ASU Tempe campus, although the goal is to scale

a system like this to integrate into any existing smart city environment. The system consists of both a unique charging solution for robotic systems, as well as a new type of battery maintenance and monitoring system for individual platforms. Project Airbud is the first testing platform for this novel battery management and charging system, and has been tested in real time on the ASU Tempe campus.

CHAPTER 2

PROBLEM STATEMENT

The focus of this project is to create a novel charging system for a fleet of unmanned aerial vehicles (UAVs). For long term robot deployment in autonomous scenarios, two of the biggest limitations are the battery life of the individual platform and routine maintenance. Each robot will only be able to operate for a set number of minutes/hours before its battery needs to be charged or swapped for a fresh pack. This power limitation imposes massive restrictions on mission planning and swarm capabilities. This is especially prevalent in the application of aerial robots, such as unmanned aerial vehicles (UAVs), that rely on active propulsion systems. In order to subvert this issue, a new type of power management system needs to be created. The outcome of this research is to develop a new charging and battery maintenance method for a confined operating area. The focus will be on allowing full UAV systems to charge their onboard power sources while not in use, and to optimize which UAV platforms need to be operational while limiting the rest to prolong battery life and limit maintenance. The biggest attribute of this system is its completely autonomous functionality, requiring no human in the loop involvement for a majority of its operations.

There are few attempted solutions for this problem; however, none are viable for long term implementation or are scalable. The most common solution is to place a human in the loop to manually swap out batteries for individual systems (Voth 2002). Although effective, this method is time consuming and it is tedious to do these operations by hand, and the operating environment for these UAV systems is not always hospitable or safe. In terms of autonomous charging, there are several solutions out on the market. Many are cost

prohibitive, with a price range between $10,000-$15,000 per platform (Al Juheshi 2017). Others, such as the wireless charging solution, take too much time and shortens the battery lifespan.

The success of this project will have several applications in modern work. The main vision of this system is for implementation in a smart city environment, to provide a fully autonomous charging system for a fleet of unmanned aerial vehicles. These UAVs would be used in the confines of the smart city for numerous purposes, including security, surveillance, traffic management, and package delivery. Incorporating a centrally controlled charging system will allow for minimum human intervention in the system at the hardware level, and will allow for easy scalability. This system is also designed to work with a scheduling algorithm and fleet management system that prioritize UAVs for missions based on battery and system health metrics, as opposed to random choice or simple distance metrics. This allows for a more efficient system in the long run, as well as minimizing downtime and maintenance.

CHAPTER 3

BACKGROUND INFORMATION

Battery Overview

In order to design a charging and battery management system properly, there needs to be a thorough understanding of how these batteries work. For most robotics and UAV applications, there are two main types of rechargeable batteries that are used, Lithium Polymer (LiPo) batteries and Nickel-metal Hydride (NiMH) batteries. LiPo batteries are composed of Lithium polymer cells, which are based on old lithium-ion and lithium-metal batteries. The primary difference is that LiPo batteries use a solid polymer electrolyte instead of a liquid lithium-salt electrolyte (Dunn 2015). Each cell has four major components: a positive electrode, a negative electrode, a separator, and the electrolyte. Just as with other lithium-ion cells, LiPos work on the principle of "intercalation and de-intercalation of lithium ions from a positive electrode material and a negative electrode material", with the liquid electrolyte providing a conductive medium (Schneider 2018). To prevent the electrodes from touching each other directly, a microporous separator is placed between the two parts, allowing only the ions and not the electrode particles to pass between sides. LiPo batteries have become the preferred power source in new electronics, especially those that are weight sensitive and require high discharge rates. However, LiPo batteries also have a shorter than average lifespan (~200 charge cycles).

Nickel-metal Hydride batteries are based off of the technologies used in the early Nickel-Cadmium batteries, utilizing a nickel oxide hydroxide base. However, the negative electrodes in NiMH batteries use a new hydrogen-absorbing alloy instead of the standard Cadmium (Kang 2006). NiMH batteries have become one of the most readily available

rechargeable batteries on the market, thanks in part of their rugged nature and low cost of production. They are available in the standard disposable battery sizes (AA, AAA, C, D, etc.) and are used portable devices built for long term use (500-1000+ cycles).

In this situation, Lithium Polymer batteries are the ideal candidate for aerial vehicles. This choice is primarily due to their lighter size, as well as high discharge rate; a complete comparison can be seen in table 1 below.

Table 1

Pro vs. Con Comparison of Lithium Polymer and Nickel Metal Hydride Batteries

LiPo Battery		**NiMH Battery**	
PRO	CON	PRO	CON
Modular sizing and shape	Short lifespan (120-250 cycles)	Longer lifespan that LiPos, 1,000+ charge cycles	Limited sizing options and configurations
Much lighter weight	Sensitive Chemistry (chance for combustion)	Less sensitive chemistry, low fire risk	Very heavy
Higher discharge rate	Special charging protocol	Simple charging requirements	Lower discharge rates
High capacity	Special storage protocol	Basic storage requirements	Lower average capacity

Each battery is defined through a rating system. The battery rating is comprised of three major components: the Discharge rate, the Capacity, and the Voltage/Cell Count (Schneider 2018). Each LiPo cell has a nominal voltage of 3.7 volts (V). Cells can be arranged in a battery pack in one of two configurations: in Series or in Parallel, denoted with the symbols ‘S’ and ‘P’. Cells arranged in Series have the voltages added together to calculate battery capacity, whereas two cells in parallel will have the same voltage. For example: a two-cell pack in series (2S) would have a nominal voltage of 7.4V, and a three-cell pack (3S) is 11.1V. A 2S2P battery would have a nominal voltage of 7.4V, but double

the current output of the previous battery (Schneider 2018). The voltage of a LiPo pack determines how fast the vehicle will move. According to Schneider (2018), voltage directly influences the RPM (rotations per minute) of the electric motor (brushless motors are rated by kV, which stands for 'RPM per Volt'). So, a brushless motor with a rating of 3,500kV will spin at 3,500 RPM for every volt applied to it. Using a 2S LiPo battery will spin that motor at around 25,900 RPM, and a 3S pack will spin the motor at 38,850 RPM. So, a greater amount of voltage correlates to a faster vehicle.

The Capacity of a LiPo battery is a measure of how much power the battery can hold, at full charge. The unit of measure is milliamp hours (mAh): how much current drain can be placed on the battery in the span of an hour. A standard iPhone has a battery capacity of approximately 3,000 mAh. It's important to note that in order to determine the charge of a battery, we look at the battery voltage, not the capacity. This is because voltage is simple to measure, whereas capacity is hard to measure accurately (Toksoz 2012). Using the voltage measurements and the discharge rate, the capacity can be calculated easily.

The last component of the battery rating system is the Discharge rate, also known as the "C" rating. The discharge rate is simply a measure of how fast a battery can be discharged safely and without causing harm to the battery. The rate is based on the capacity of the battery, which denotes the "C." For example, a battery with a 5C discharge rating means it can discharge at up to 5 times the battery capacity; for a 50Amp battery that's a max discharge rating of 250Amps. The resulting number is the maximum sustained load you can safely put on the battery. Going higher than that will result in, at best, the degradation of the battery at a faster than normal pace. At worst, it could burst into flames. Modern LiPo batteries have two discharge ratings, a continuous rating (which has been

discussed thus far), and a burst rating. A burst rating works the same way, except it is applicable for 10 second bursts, not continuous discharging. This is taken into consideration for short spikes in power consumption, such as a vehicle acceleration. Burst ratings are almost always higher than the continuous ratings. For most applications, a 20C-25C rating for a battery is satisfactory.

Charging a LiPo battery

Due to its special construction of multiple lithium polymer cells, a LiPo battery cannot be charged using just an adequate power source; instead it requires a specialized power source and control system. LiPo batteries are charged using a Constant Current/Constant voltage system (CC/CV); basically, the charger delivers a set current (charge rate) to the battery until it reaches its peak voltage, usually 4.2 volts, and will then maintain the voltage while reducing the current. This is in comparison to NiMH batteries, which use a pulse charring method.

LiPo charging also requires another component: a balancing circuit/system. Balancing is the act of equalizing the voltage of each cell in a battery pack. Each cell is balanced in order to ensure they all discharge at the same rate and same amount. This contributes not only to battery health, but safety as well. Most LiPo batteries need to be charged rather slowly, compared to NiMH or NiCd batteries. While it is routine to charge a 3000mAh NiMH battery at four or five amps, a LiPo battery of the same capacity should be charged at no more than three amps. Just as the C Rating of a battery determines what the safe continuous discharge of the battery is, there is a C Rating for charging as well. For the vast majority of LiPos, the Charge Rate is 1C. The equation works the same way as the

previous discharge rating, where 1000mAh = 1A. So, for a 3000mAh battery, we would want to charge at 3A, for a 5000mAh LiPo, we should set the charger at 5A, and for a 4500mAh pack, 4.5A is the correct charge rate.

Discharging and Use

In general, a fully charged LiPo battery has a voltage of 4.2V per cell. Anything over this will cause the battery damage. As a LiPo battery is discharged, each of the internal cells is discharged at the same rate, providing a steady DC power output. It is very critical that the cells do not drop below 3V, as this causes a usually permanent degradation of the cell's ability to absorb and retain a charge. In light of this, most manufacturers have taken to putting a Low Voltage Cutoff (LVC) in their control systems. The LVC detects the voltage of the battery, and divides that voltage by the cell count of the battery. So it would see a fully charged 2S LiPo as 8.4V, or 4.2V per cell.

This is where the advantage of balancing comes in. Because the speed control does not read off the balance tap, it cannot know the exact voltages of each cell within the battery. The speed control can only assume that the cells of the battery are all equal. Most LVCs cut off around 3.2V per cell, or at least signal a low voltage warning; this ensures the battery health is maintained. If the cells of a battery are not equal, the LVC may not engage when necessary; for example, For the two-cell (2S) example battery, that would be 6.4V. But if the battery isn't balanced, it's possible for the total voltage to be above the cutoff threshold, yet still have a cell below the 3.0V danger zone. One cell could be 3.9V, while the other could be a 2.8V. That's a total of 6.7V, which means the cut-off would not

engage. The vehicle would continue to operate, allowing the further degradation of the battery.

Unlike standard disposable batteries or the NiMH batteries, LiPo batteries are a little more delicate and need to be handled with care. This includes using them (discharge); if not handled correctly, a LiPo can catch fire (Schneider 2018). This is due to the battery chemistry. Lithium-Polymer batteries contain lithium. Lithium is an alkali metal, meaning it combusts when reacting with water. Lithium also combusts when reacting with oxygen, but only at high temperatures. The process of using the battery, especially at high discharge rates as with robotics, causes there to be excess atoms of Oxygen and excess atoms of Lithium on either end (anode/cathode) of the battery (Li 2009). This causes Lithium Oxide (Li2O) to build up on the anode or cathode. Lithium Oxide is basically corrosion, similar to rust, which is iron oxide (Schneider 2018). The Li2O causes the internal resistance of the battery to increase. This causes the battery pack to heat up even further during normal use, increasing the operating temperature. Heat causes the excess oxygen to build up more and more. Eventually the LiPo pack begins to swell (due to the oxygen gas build up). When a LiPo battery pack has become swollen, it is a clear indicator to stop using the battery; at this point it has been damaged internally and will not perform as expected. If a LiPo battery is punctured, it will also swell, and potentially ignite (Dunn 2015). This is because the lithium reacts with the humidity in the atmosphere and heats up the battery. This heat excites the unstable bonds, which break, releasing energy in the form of heat. The entire process of building up that lithium oxide usually takes around 300-400 charge/discharge cycles to reach a tipping point. That's a typical lifetime of a LiPo battery. But when we heat the batteries up during a run, or discharge them lower than 3.0 volts per cell, or physically

damage them in any way, or allow water to enter the batteries (and I mean inside the foil wrapping), it reduces the life of the battery, and hastens the buildup of Li2O.

Storage

For optimal longevity, a LiPo battery should be stored at room temperature with a charge of 3.8V per cell. Leaving LiPo batteries at full charge for as little as a week can cause damage to the inside of the batteries. This is the same as laptop batteries, which should never be stored with a battery level over 60%. LiPo batteries should never be stored on board a vehicle (without proper isolation), and should be kept out of the sunlight.

CHAPTER 4

LITERATURE REVIEW

There are currently a few different implementations that exist, in both academia and in industry, for quick turnaround Unmanned Aerial Vehicle (UAV) battery charging/management. On the industry side there are a couple of small companies that are trying to focus on autonomous charging for UAVs. WiBotic, a Seattle-based startup, has created an integrated wireless charging pad for a single vehicle. Each UAV has an onboard receiver that is able to receive power from the charging pad's transmitter through wireless transmission. The PowerPad provides 100 watts of power and can provide a full charge for a 20-minute UAV battery in approximately two hours (Kit 2018). Figure 1 below is a graph of efficiency of their wireless charging capabilities. Although their efficiency is greater than standard wireless charging platforms, the limitations with distance and coil size still exist.

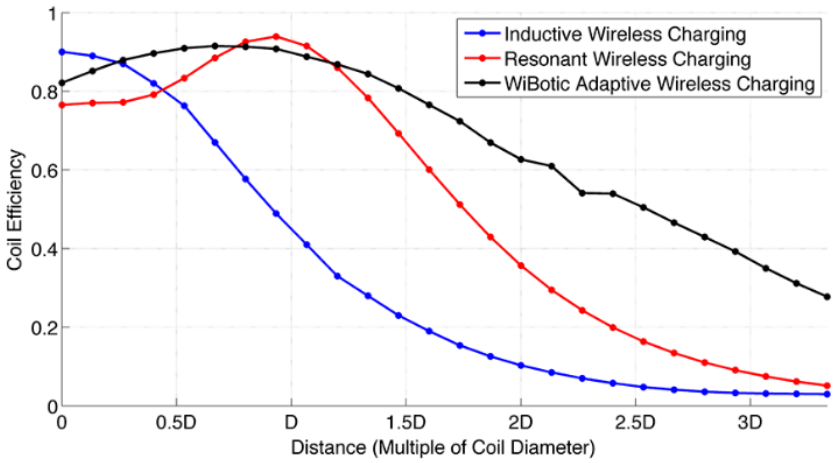


Figure 1: Charging efficiencies of different wireless charging methods (WiBotic 2018)

SkySense has created one of the first contact charging systems for UAVs, and is selling prototypes of both indoor and outdoor systems. The system is comprised of two parts, a

base station which is a large contact, and a retrofit kit for the board. The base station contains a moving part that connects to a part of the UAV for power transmission. This system costs $15,000 and will charge a 20-minute flight time battery in about an hour (Skysense 2018).

Outside of the UAV industry, numerous companies have been working on charging solutions for large battery packs, primarily in the electric vehicle industry. Qualcomm has recently announced their focus on wireless electric vehicle charging. Dubbed "Halo," this system uses wireless transmitters to turn a parking spot into a wireless charging system, for any electric car equipped with a wireless power receiver. The base pad boasts a power transfer rate of 22kW, with a 90% efficiency rating (Borroni-Bird 2013). Other automotive OEMs are also considering equipping new cars with wireless charging receivers, as well as creating retrofittable components to bring this technology to existing electric vehicles.

On the research side in academia there has also been an interest in finding better ways to charge/manage battery systems for UAVs, as well as work on improving LiPo battery technologies and battery management systems. The Aerospace Controls Laboratory led by Dr. Johnathan How at the Massachusetts Institute of Technology has been working on a project for automated battery replacement for UAVs. Their focus has been on charging batteries offboard the UAVs and using a mechanical system to swap the batteries in place when the UAV has landed (Ure 2015). Currently, this project is focusing on small micro-UAVs, with single cell LiPo batteries that can be quickly hot swapped without much mechanical effort (Bethke 2009). A diagram of the UAV platform is shown in Figure 2.

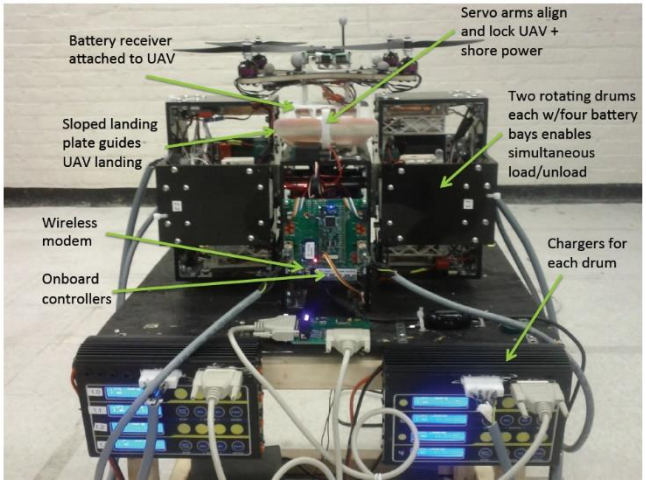


Figure 2: Diagram of MIT's micro UAV battery swap station (Ure 2015).

A team at the University of Tokyo has also prototyped a similar system, focusing on swapping the spent LiPo battery with a fresh pack while the UAV is docked in the landing station (Fujii 2016). Their implementation uses a motor powered mechanical gripper to grasp and remove the battery pack from underneath the UAV frame. The battery pack is unique because it slides into a holding case below the UAV main frame, with two contact points that transfer power to the power distribution board. Although this design makes it easier to access and swap the battery, it wastes a large amount of space and adds extra mass to the platform. A picture of this is shown in Figure 3.

Figure 3: Custom UAV landing pad at University of Tokyo (Fujii 2016).

In addition to full systems that allow for battery swap or contact charging, a large body of the work done on this problem has been in developing better power solutions for UAVs. This includes newer forms of power storage and more efficient on-board power management systems; significant progress has also been made on converting UAV components to low-power consumption models, to decrease overall power usage. A group at the University of Pisa has made significant improvements to battery management system (BMS) technologies with their creation of a hierarchical platform that allows for the modular charge balancing of LiPo batteries in series (Baronti 2011). The biggest influence from this work is a novel metrics and evaluation method for LiPo balance and health data, which can "provide valuable comparisons and information about the efficiency and the balancing time of different circuitries" (Baronti 2011). This is paramount in work relating to LiPo health monitoring, as it allows for comparisons in minute battery state changes, which can lead to detecting health-risking anomalies. More recently, electrical engineers from Zhejiang University have developed a mixed-method for charging LiPo batteries, utilizing both packing charging and cell charging mode (Zhou 2014). This algorithm allows for a LiPo battery to be uniformly charged by one source (via the pack charging standard) while also optimizing balanced charge levels in real time (the cell charging mode). This method allows for faster charging of LiPo batteries while maintaining balanced cells across the pack, which is crucial for battery safety. Lastly, a team at MIT has developed a strategy to help design higher rate battery electrodes for lithium manganese batteries; these designs allow for safe use of high-discharge and charge rate batteries, as well as improved energy density (Kang 2006). This work is crucial for the development of large-capacity LiPo batteries that have to discharge at an instantaneous rate of over 100C, as it will prevent the

battery electrodes from failing, which leads to battery health decline or high temperature spikes (leading to combustion).

CHAPTER 5

SYSTEM OVERVIEW

In order to create a more effective system for charging and maintaining batteries for a swarm of Unmanned Aerial Vehicles (UAVs), a new system was designed from the ground up. This system utilizes contact charging between a grounded base station and the UAV system's onboard battery. This project included both Hardware and Software components, for both the base station system as well as the UAV on board battery management and distribution system.

Hardware Components

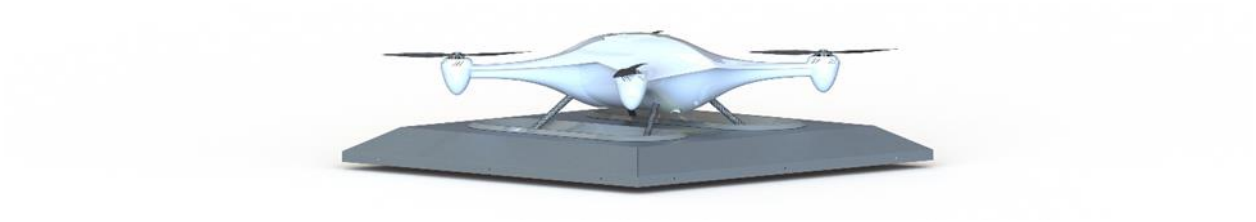


Figure 4: Full CAD rendering of final base station and UAV design

There are two major hardware components for this project. The first component is the base station, which acts as a landing pad and charging pad for any pre-equipped UAV system. This base station, pictured in Figure 4, is a self-contained system that is designed to charge any system using standard rechargeable batteries. Inside its casing is a full power supply system that provides a constant current, constant voltage CC/CV power output. This output is connected to two brass rails, which serve as the contact point between the base station and a UAV system. The base station also contains an automotive-grade microcontroller that handles the control of the charging system, as well as internal diagnostics and environmental tracking tools. This serves two purposes: to maintain a safe

and efficient charging system for UAVs, and to monitor for any faults in the system or external conditions that can pose danger for the base station or UAV systems.

The other major hardware component is the on-board UAV battery hardware. In order for a power system to work with contact charging, the battery needs to be equipped with certain additional hardware. First and foremost, the battery system needs to be standardized (size and capacity). Second, the on-board battery requires a battery management system (BMS) board that will regulate the charging of the individual cells in the battery, very similar to a LiPo balancing circuit. This BMS board receives power input directly from the base station, via charging contacts on the UAV's landing gear, and distributes the power to individual cell blocks for an even charge rate across the battery. The BMS is programmable and is set to charge to a certain point. In our work, we have created a standard battery for use on the AirBud UAVs, and have built a custom battery system as well as sources a standard Consumer Off the Shelf (COTS) product. A model of the UAV power system and battery is shown in Figure 5.

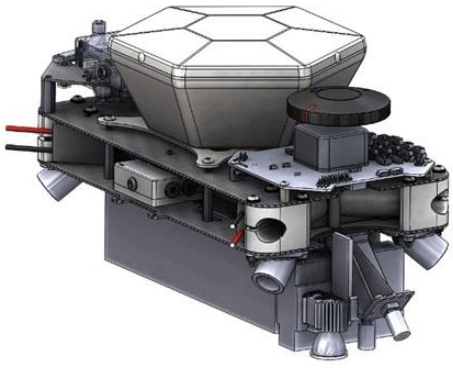


Figure 5: UAV power system CAD model.

Software Components

There are three major software components for this project. The first is the software that controls the base station; the second is the software on board each UAV that manages the onboard battery system and monitors system health; the third is the charging management system which is a cloud-based API.

Each base station is equipped with two small computing modules. The first, a Linux-based embedded computer, is used to manage and run the power supply system. The software running on this board monitors the base station for a UAV system, and verifies the charging parameters for each UAV via the fleet management system. The software then activates the power supply system and allows the UAV to charge its battery through the contact rails embedded inside the charging pad. Additionally, the software monitors the power output of the system to verify proper charging conditions. A secondary microcontroller is placed in each base station; this board gather environmental data from both inside and outside the base station, and the software monitors for any unsafe conditions (high internal temperature, harsh rain, etc.).

Onboard each UAV system is an Intel-based computer, which runs the control system and mission planner for each independent platform. A piece of the software running on this board is dedicated to monitoring and managing the UAV battery system. This includes setting the max charge voltage for the battery, as well as interpreting battery health based on data from the battery management system. This compute module is also in constant communication with the fleet management system, constantly relaying battery levels and heath metrics.

The last piece of software is the charging management system, based in the cloud server along with the fleet management system. This software has two main duties: determining the charged battery levels for each UAV system, and determining when UAV systems need maintenance or relocation based on battery metrics, current demand, and overall system health factors. This software runs as a microservice as a part of the fleet management system, and is accessible via a custom API designed for this distributed UAV platform.

CHAPTER 6

HARDWARE SYSTEM

UAV Battery System

Control system & diagram. The UAV systems that have been built to work with this contact charging pad have been designed with a custom power system, composed of off the shelf hardware. The drone power system includes the battery pack, the battery management system (BMS), the TerminateAir flight termination system, the UAV control carrier board, and the charging rails that supply power to the drone (embedded in the landing gear). Power onboard the UAV is supplied by a 6S (nominal 22.2V) 18000-mAh lithium ion battery with an integrated BMS/balancing system. A LiPo battery has been chosen due primarily to its low weight and high discharge rate, a necessary feature for an aerial robot depending on high powered motors to keep it afloat. A more detailed analysis of the batteries chosen will be in the next section. A full-wave bridge rectifier consisting of four Schottky diodes is used to isolate the battery from the charging rails while still providing a means to charge the battery from the contact rails. The rectifier also allows the UAV to utilize the charging base station in either a front-facing or rear-facing orientation. The output of the battery is connected to a TerminateAir power controller. The TerminateAir serves to disconnect power to the UAV system in the event of an emergency situation, such as a loss of flight control or a failure of the propeller motors. The TerminateAir also has an internal BEC (battery-elimination circuit) which is used to power its companion controller, the SafeAir. The SafeAir module monitors the motion and performance of the UAV while in flight, and serves to trigger a parachute deployment and a deactivation of the TerminateAir power output in the event of an emergency situation.

Power to these two boards is critical, so they are isolated from the rest of the electrical systems on the UAV. The battery voltage supply from the TerminateAir is connected to a SpektreWorks carrier board which is controlled by a Pixhawk flight controller. The carrier board contains onboard DC-DC converters which supply the necessary voltages for the Pixhawk and some peripheral components. The carrier board also serves as a battery-voltage power distribution bus and PWM-signal controller for the four ESCs (electronic speed controls) which drive the BLDC (brushless DC) propeller drive motors. The battery voltage supply from the TerminateAir is also connected to two external 3-amp DC-DC buck converters. One such converter is used to supply a 5V rail to an Linux-based onboard computer. The onboard computer receives image information from mounted cameras and provides autonomous navigation instructions to the PixHawk flight controller. The second external buck converter provides a 4.2V rail to the high-intensity spotlight which is mounted to the front of the UAV. The light is controlled by a logic-level control signal from the onboard computer. A detailed diagram of the full UAV power system is shown in Figure 6. Individual subsystem schematics as well as a high-level diagram are shown in Appendix A.

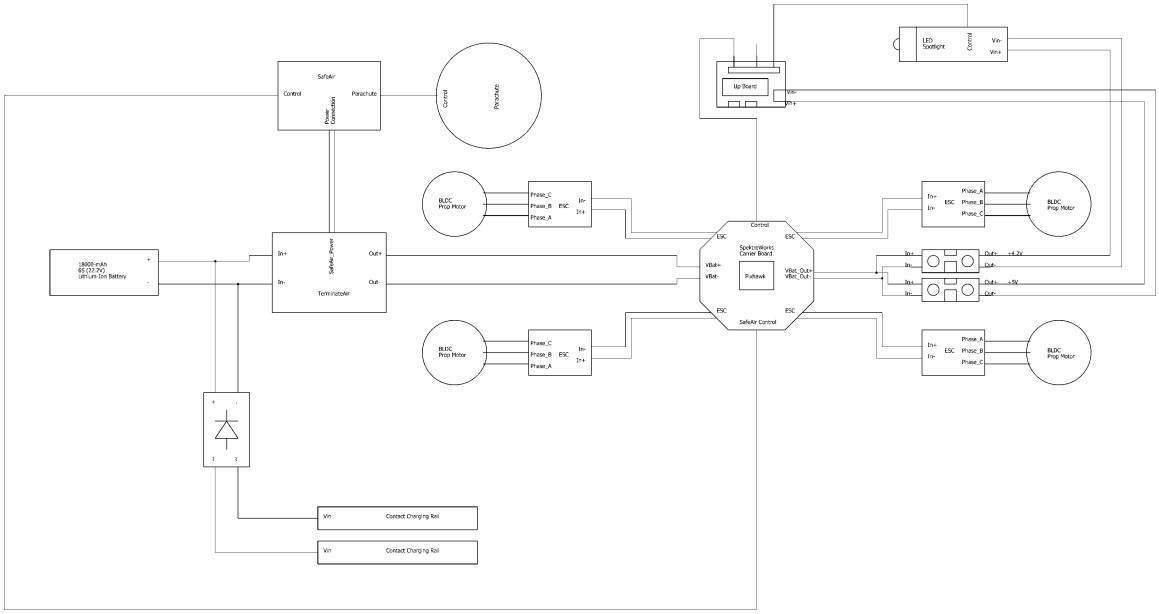


Figure 6: Schematic of UAV power and charging system.

A detailed power budget for the UAV system is shown below. This is based on averaged flight data from testing, as has been verified against manufacturer guidelines.

Table 2

UAV Power Budget/Schedule of Loads

Load:	**Voltage Requirement:**	**Power Source:**	**Approximate Max. Power:**
Propeller Motors	20-25.2V	SpektreWorks	3024 W
Computer Board	5V	3A DC-DC Converter	15 W
Spotlight	4.2V	3A DC-DC Converter	10.5 W
Pixhawk	5V	SpektreWorks	10 W
SafeAir	15V	TerminateAir BEC	10 W
Battery (Charging)	20-25.2V	Base Station	302 W

COTS battery. In order to quickly develop the custom UAV system and the charging station, an off the shelf (COTs) battery was chosen as the first power supply. There were two major requirements for the battery that was chosen: it had to have at least

22V of nominal voltage, and needed to have a built-in battery management system. The latter is due to the need to manage LiPo cell balancing onboard the UAV as opposed to the base station, which would require more complex mechanization and wiring. Due to these constraints, a 6S1P battery was the ideal candidate, with a nominal voltage of 22.2V at a 15C discharge rate. We decided to purchase the Tattu Plus 2.0 18000mAh battery pack. Table 3 below shows detailed specifications for the battery.

Table 3

Tattu Plus 2.0 Battery Specifications and Measurements

Minimum Capacity	18000 mAh
Configuration	6S1P / 22.2 V / 6Cell
Discharge Rate	15C
Max Burst discharge rate	30C
Net Weight	2270g ± 20g
Dimensions	205.5mm L x 94.2m,m W x 79.5mm H
Discharge Plug	AS150+XT150
Charge Plug	XT90

One of the major benefits of this board is that it has a built-in battery management board, which automatically balances the LiPo cell voltage inside the battery. This means all the battery needs to be charged is a connection to a constant current constant voltage power source, and the rest is handled onboard. The Tattu battery also has several safety features that makes it a prime candidate. First it has dual levels of charge protection built I; this guarantees the charging circuit will be cut off in case of improper power supply or potential damage to the battery. This battery also is ruggedized against physical trauma (a must for a UAV battery), and is able to be stored in above-average temperatures (ranging from 158F to 176F). Lastly, the 2.0 version of this battery gives us access to the onboard BMS, allowing us to set the voltage cutoff and charge rates, and monitor individual cell

voltage levels, which allows us to gauge the battery health. This off the shelf battery also fulfills the ISO90001 and ISO14001 safety standards, which are necessary for any vehicles operating in close proximity to unprotected individuals.

Custom battery design. The custom-constructed 18650-based drone battery utilizes a 6P/6S configuration of high-capacity (3000mAh) 185650 cells, for example, LG HG2 cells. The combined nominal voltage designated by the 6S configuration is 22.2V and the combined charge capacity is 18000mAh.

The batteries are monitored and balanced by a 6S BMS with maximum current capacity of at least 100A. Example options of such a BMS include the BesTech HCX-D328. The BMS is configured with balance leads connected to each 6P cell block for charging and discharging balancing as well as health monitoring of the individual cell blocks.

Physical construction of the battery pack optionally consists of a plastic or polymer-based carrier frame for the individual 18650s, or alternatively a shrink-wrap and high-temperature fiberboard construction. Cells are ideally connected using tack-welded nickel plate strapping, or alternatively direct-soldered copper-braid. A picture of the custom battery is shown in Figure 7.

Figure 7: Custom LiPo battery with BMS board

Base Station System

Mechanical design & shell. The base station designed for this project is enclosed in a large trapezoidal metal shell, approximately one meter on a side. The inside of this shell contains the power generation subsystem, as well as the control system. The inside is completely sealed off from the outer environment, with the only part connecting the two being the contact points. These contact points are located in the grooves on top side of the system, which is where the UAV platform lands when docking with the charging system.

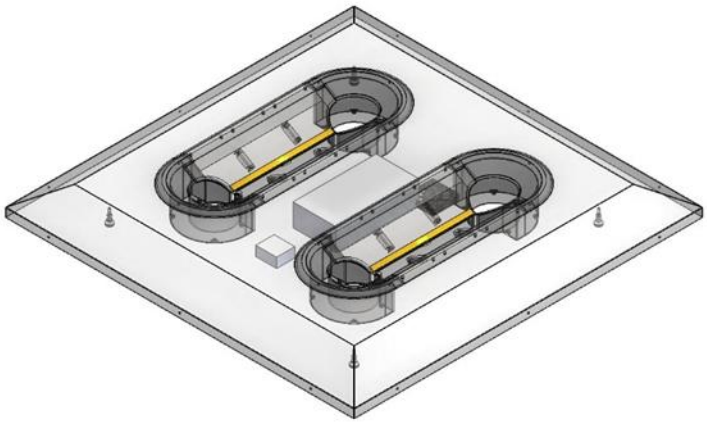


Figure 8: Base station 50% transparency CAD render

Before the base station was physically built, a prototype design was created using computer aided drafting software. In this software, we were able to build the foundation of the base station, and modify the parameters to meet the requirements. This was crucial for making sure the system would be large enough to accommodate the operating envelope of a UAV platform. The final CAD designs of the base station are shown below. In the center of the top piece, two large tapered slots were designed to accommodate standardized mating landing gear. They slope inwards to allow for a larger landing area for the UAV; as the UAV lands on the base station, the RTK precision GPS is used to align the operating envelope over the base station. As the UAV begins its controlled deceleration (as thrust is

reduced), the landing gear makes contact with the covered slots. Due to gravity, the UAV's landing gear will slide down the tapered sides and lock in place at the bottom of the slot. This ensures the contact points on the base station and UAV landing gear will line up perfectly. The slots are covered in a plastic material, in the shape of cones, which serves as a large insulating layer, as well as prevents the two rails from ever shorting. Having this made of plastic ensures the UAV landing gear and the outer shell are not scratched during a landing procedure. A model of the slots is shown in Figure 9.

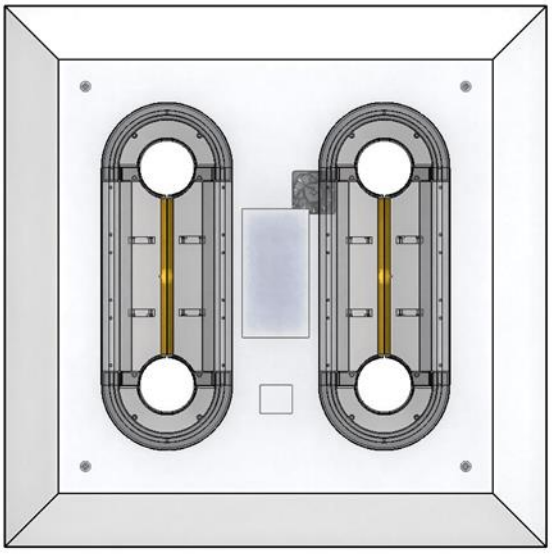


Figure 9: Top view of base station tapered slots with metal contacts

The electrical systems of the base station were designed in parallel with the shell designs. Using SolidWorks, we laid out all of the electronic components inside the base station, determining positioning and placing of each component. This let us model all of the wiring and connections in the system, determining the exact amount of wiring needed for this assembly.

For the prototype of the base station that was built, we decided to use aluminum for the outer shell. This is due to the fact that aluminum is a strong, cheap metal that is easy to work with. It is also an industry standard for outdoor electronics enclosures.

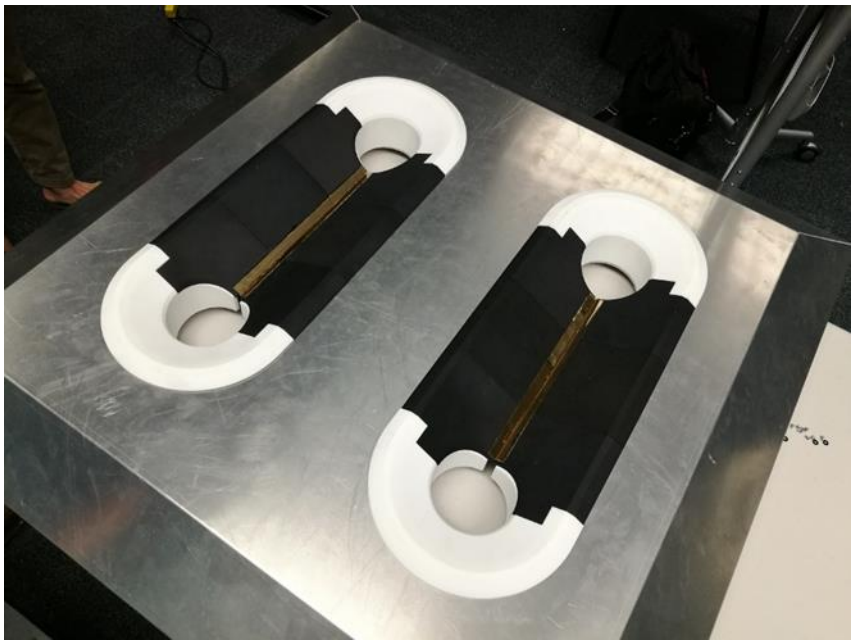


Figure 10: Base station metal prototype with 3D printed slots

The metal was cut using a water jet system, and the top grooves for the landing points were bent using an industrial metal bender. The aluminum will be anodized, allowing it to have a water-protective coating, which is ideal for outdoor use. The slot cones were 3D printed using PLA plastic; this allowed for a custom fit that aligns well with the metal slots. The cones were split into several distinct pieces, which were bolted together during assembly. A diagram of the UAV landed on top of the base station can be seen above in Figure 10.

One of the main decisions for this prototype was to determine which conductor to use for the contact points on the base station. Candidates included aluminum, gold-plated metal, nickel-plated metal, pure copper, brass, and carbon contact points. Several of these metals were tested for their longevity, efficiency as a conductor, and resistance in the end,

brass was chosen due to its low internal resistance and durability, especially to oxidation. The brass is attached to the top of the plastic slot cones; two pieces of brass are connected together in a “V” shape, and sit at the bottom of the slot. The power delivery wires from the DC-DC Converter are soldered to the bottom of the brass and run through the 3D printed cones.

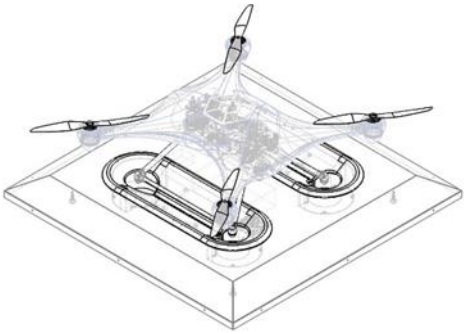


Figure 11: Diagram of UAV platform docking with base station contact points

Lastly, one of the important designs for the inside of the base station was managing the cooling system for the base station. Due to the rooftop location of these platforms, and the high temperature levels in Arizona during the summertime, thermal considerations were of high priority during development. In order to prevent overheating of electronics components during high ambient temperature and high-power consumption conditions, an active cooling system was incorporate into the base station. This system consisted of two fans, one dedicated to the DC-DC converter, and the other for the rest of the system. Intake and exhaust vents were strategically placed around the perimeter of the shell. Thermal testing was conducted on the system during normal operations, and industrial-grade components were chosen to verify continued operation.

Control system and diagram. The base station charging system is comprised of the following major electronic components: a 36-volt 400-watt power supply, a constant-current/constant-voltage (CC/CV) buck converter, a pair of brass contact rails, and a control/measurement system. The 36-volt 400-watt power supply serves to convert the 100-130V AC voltage supply to the charging base station into a constant 36-volt supply rail for the charging equipment. The nominal charging power delivered to the drone is 302 watts and the DC-DC converter is estimated to be 90% efficient, so a total power consumption of 334 watts is expected to be placed on the power supply. This is 84% of the specified power supply's capacity, and as such is well within its rated operating conditions. A diagram of the power management portion of the system is shown in Figure 12.

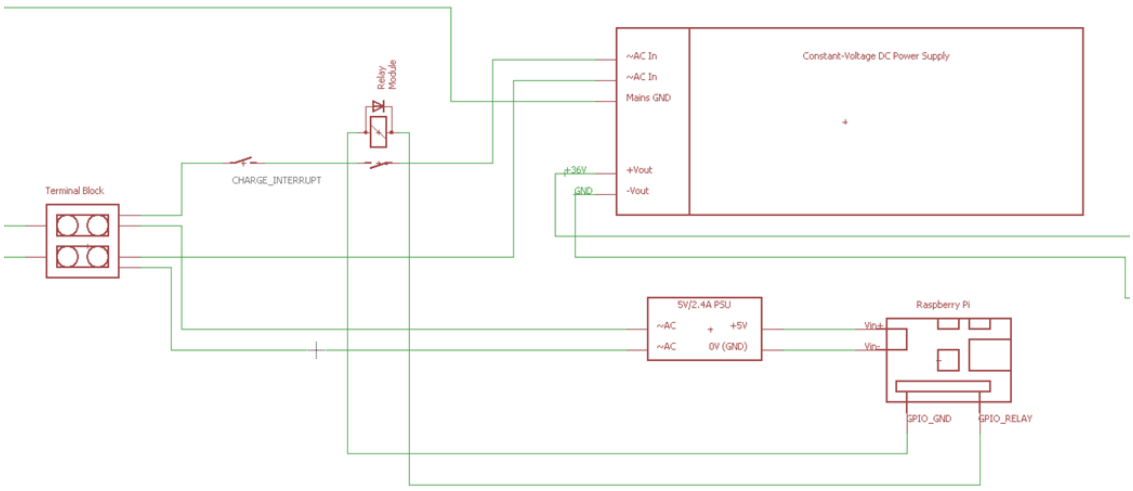


Figure 12: Electrical schematic of base station power management system

The CC/CV buck converter accepts and input range of up to 42VDC and outputs a maximum constant current of 12 amps and a maximum constant voltage of 25.2V. This voltage has been selected in order to charge the 6S lithium-polymer battery pack onboard the drone to a nominal maximum voltage of 4.1V/cell. This voltage assumes a forward voltage drop in the Schottky-barrier rectifier onboard the drone of 0.6V. The CC/CV

converter is supplied with a constant 36V voltage supply form the power supply, and the maximum power rating on the CC/CV buck converter is 400W. The brass contact rails are mounted on the top of the base station in recessed slots, separated by a fixed distance corresponding to the width of the charging rails on the drone. The flat surfaces are placed at a 135-degree angle to one another in order to allow for a wide landing angle for the drone. Contact pressure is maintained by the gravitational force exerted by the drone on the contact interface.

The control system consists of a Raspberry Pi, a wireless communication interface, a separate 5-volt power supply, and a 250V/10A logic-level relay to interrupt power to the charging equipment. The control system utilizes the relay to interrupt power to the 36-volt 400-watt power supply when the drone is not present at the base station for safety and to eliminate arcing at the contact rail interface. By communicating actively with the drone control system via the wireless communication interface, the control system withholds power from the charging circuit until the drone has securely landed. Once the landing has been confirmed, the power supply closes the relay and activates the charging circuitry.

Additionally, the charging voltage and current are monitored using an onboard analog measuring system within the base station. Base station-side voltage is monitored via a fixed voltage divider connected directly to a GPIO input on the Raspberry Pi and is compared to the measured battery voltage onboard the drone in order to assess the health and resistance of the brass-to-brass rail contacts. The current supplied to the drone is monitored using a 10-milliohm shunt resistor whose terminals are connected to a finite-gain differential amplifier. The differential amplifier's gain is set nominally to 30 in order to produce a measured GPIO voltage of 3.6V when charging current through the shunt

resistor is 12 amps (the maximum nominal voltage). Charging voltage is maintained for the entirety of the drone's presence on the base station in order to maintain pass-through power supply to active components onboard the drone. Power to the charging circuit is interrupted shortly before liftoff at the request of the drone's onboard control system.

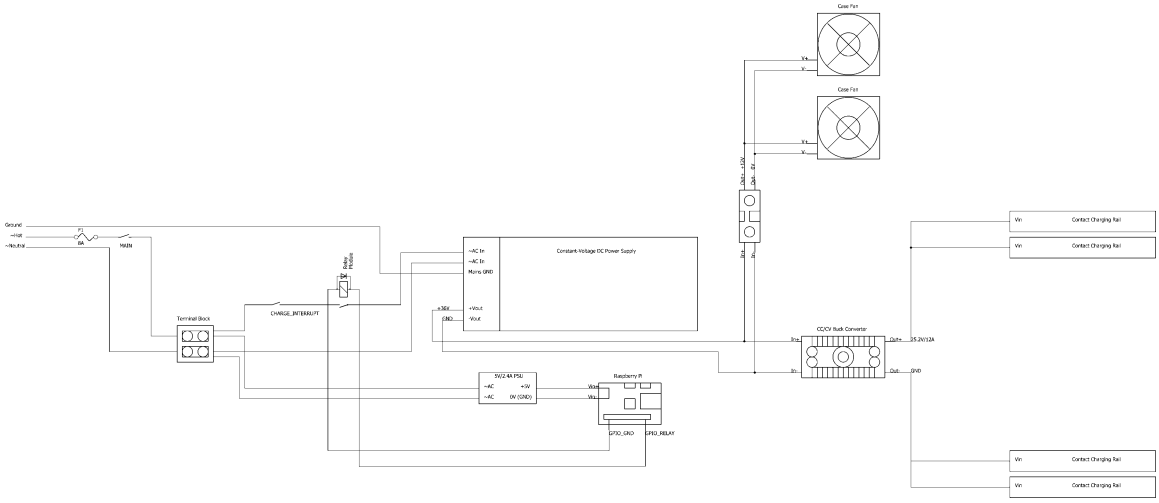


Figure 13: Schematic of entire base station electrical system

Software management & protocol. The charging system consists of a charge-control system on the base station as well as a wirelessly-connected charge and contact-monitoring system onboard the drone. Battery voltage is measured onboard the drone and reported to the charge control algorithm via a high-impedance, high-voltage analog-to-digital converter. In the event of a battery fault or other non-ideal charging condition, the onboard charge control system may request that the base station interrupt battery charging at any time. Furthermore, as dictated by a drone-distribution and utilization algorithm, charging may optionally be withheld such that the drone's state-of-charge is maintained at a nominal storage-voltage for long-term storage.

Onboard the base station, a wirelessly-connected Raspberry Pi is used to control the operation of the charging circuit via a binary logic relay. The Raspberry Pi's internal

ADC (analog-to-digital converter) is also employed to monitor the charging current via a 10-milliohm shunt resistor and a finite-gain differential amplifier.

The relay controls the delivery of AC power to a 36V, 400W power supply unit which provides a fixed 36V rail to the remaining charging circuitry. The 36V rail subsequently supplies a constant-current, constant-voltage DC-DC buck converter which ultimately regulates the maximum charge voltage of the battery as well as the current at which it is charged.

Separate from the charging control system, there is also an internal monitoring system that has been designed for the final version of the base station, and is currently being tested inside a prototype. The goal of this system is to monitor the status of the charging system, as well as the internal and external environment of the base station. This is done via a collection of IoT sensors that are polled several times a minute to gather specific information. The first group of sensors consists of voltage and current sensors: these are used to track the power flowing through the base station, and ultimately the output being delivered to the UAV via the contact rails. The second group of sensors are temperature and motion sensors inside the base station; these monitor the internal conditions of the system, monitoring for overheating or extreme vibration (signs of component failure or overuse). The last group of sensors include moisture and external temperature and light sensors; these are used to detect the outside environmental conditions, include sunshine and rain. These can be used to determine the operating environment of the station, and whether or not it needs to be covered/disables (such as in the case of rain).

Charging types. Depending on the distribution of drone utilization within the fleet of drones, the charging system can be configured to apply one of the following five different charging profiles. These profiles are ranked from highest-performance and shortest battery lifespan-promoting to lowest-performance and longest battery lifespan-promoting.

Floating consumer-specification charging (profile 1). This charging profile is recommended for drones which are predicted to be in use for long-duration or power-demanding flights in close temporal proximity (< 1 week) to the time of charging, and which are subject to high parasitic loads (i.e. flight control systems, onboard computing, lights, etc.). Charging profile 1 charges the battery to the consumer specification of 4.20V per cell and provides an indefinite constant-voltage (CV) float voltage to the battery pack. This CV float voltage assures that battery discharging and cycling does not occur due to parasitic loads onboard the drone.

Charging profile 1 begins charging in constant-current mode at the maximum charge rate supported by the charger and/or battery system. Charging is continued until the battery pack voltage reaches 4.20V per cell, at which point the charger enters constant-voltage mode. In CV mode, the battery is saturation-charged, and is subsequently held at the CV point of 4.20V per cell for an indefinite period of time. Charging is only interrupted when the drone prepares for takeoff or when a different charging profile is selected.

Windowed consumer-specification charging (profile 2). This charging profile is recommended for drones which are predicted to be in use for long-duration or power-demanding flights in close temporal proximity (0-2 weeks) to the time of charging, but which are not subject to high parasitic loads (i.e. flight control systems, onboard

computing, lights, etc.). Unlike charging profile 1, charging profile 2 does not maintain a float-charge voltage at 4.20V per cell, which over time may cause battery degradation. Rather, it engages saturation charging to 4.2V per cell, followed by windowed re-charging at 4.1V per cell.

Charging profile 2 begins charging in constant-current mode at the maximum charge rate supported by the charger and/or battery system. Charging is continued until the battery pack voltage reaches 4.20V per cell, at which point the charger enters constant-voltage mode. In CV mode, the battery is saturation-charged until the supply current drops below 3% of full charging rate, at which time the charging circuit is disabled and battery voltage is allowed to equilibrate. Should battery voltage fall below 4.10V in this profile (whether due to parasitic loading or otherwise), charging is re-enabled and the saturation charge is again applied as before.

Military-specification charging (profile 3). This charging profile is recommended for drones which are predicted to be in use for moderate-length flights which are in the near- to far- future (0-4 weeks). Charging profile 3 charges the battery to the military specification of 3.90V per cell and provides an indefinite constant-voltage (CV) float voltage to the battery pack. This CV float voltage assures that battery discharging and cycling does not occur due to patristic loads onboard the drone, and is of minimal hazard to battery lifespan due to the lower-than-maximum battery charge voltage.

Charging profile 3 begins charging in constant-current mode at the maximum charge rate supported by the charger and/or battery system. To improve battery lifespan, charging current may optionally be reduced to 0.5C if charging hardware and operating timetable constraints allow this. Charging is continued until the battery pack voltage

reaches 3.90V per cell, at which point the charger enters constant-voltage mode. In CV mode, the battery is saturation-charged, and is subsequently held at the CV point of 3.90V per cell for an indefinite period of time. Charging is only interrupted when the drone prepares for takeoff or when a different charging profile is selected.

Floating long-term storage (profile 4). This charging profile is recommended for drones which are not anticipated to fly and are to be kept in storage for long periods of time. Short-duration flights are permissible though not recommended. Floating-mode charging is recommended for drones which are subject to high parasitic loads (i.e. flight control systems, onboard computing, lights, etc.).

Charging profile 4 will not enable charging at all if the drone's battery voltage is above 3.7V per cell. Instead, it will wait for one of the following loads to discharge the battery to below 3.7V per cell: onboard parasitic loads/electronics, subsequent flights, or intentional non-flight propeller running for discharging. When battery voltage is detected to be below 3.7V per cell, the charger will engage at the rated current of the battery and/or charging system. When the battery achieves a cell voltage of 3.7V per cell, the charger enters CV mode and remains in this mode indefinitely. Charging is only interrupted when the drone prepares for takeoff or when a different charging profile is selected.

Windowed (top-up) long-term storage (profile 5). This charging profile is recommended for drones which are not anticipated to fly and are to be kept in storage for long periods of time. Due to the lower-bound of the windowed charging of 3.5V, flying is not permissible under this charging mode. Windowed-mode charging is recommended for drones which are subject to minimal parasitic loads.

Charging profile 5 will not enable charging at all if the drone's battery voltage is above 3.7V per cell. Instead, it will wait for one of the following loads to discharge the battery to below 3.7V per cell: onboard parasitic loads/electronics, subsequent flights, or intentional non-flight propeller running for discharging. When battery voltage is detected to be below 3.7V per cell, the charger will engage at the rated current of the battery and/or charging system. When the battery achieves a cell voltage of 3.7V per cell, the charger will interrupt. Voltage monitoring will continue to eliminate risk of undercharging, and if cell voltage drops below 3.5V per cell, charging will be re-enabled until cell voltage again reaches 3.7V per cell.

Pass-through charging (detailed). Pass-through charging occurs when current is drawn from a battery system by onboard loads while battery charging is taking place. In the charging configurations described, both the battery charging power supply and the pass-through loads are connected directly to the battery terminals. As a result, the current supplied by the battery charging power supply is divided between charging the battery and supplying the loads when the battery is charging.

Float-voltage charging and voltage-windowed charging profiles are supported by the charging system and both profiles can be used with pass-through charging. Float-voltage charging should be preferred when pass-through charging is used in order to reduce cycling-load on the battery. When windowed-charging is used, pass-through charging may result in repeated cycling of the battery between the start- and end-charge voltages, and as such it is less desirable. For prolonged use of float-voltage charging, float voltage should be set slightly below 100% state-of-charge voltage for the battery in order to reduce degradation of the battery over time.

During either charging profile, battery charging current is set to a fixed maximum value. If the total current drawn by the pass-through loads exceeds the maximum charging current, then the battery will begin to discharge in order to provide the remaining current to the load. In contrast, if the total current drawn by the pass-through loads does not exceed the fixed maximum charge current, then the remaining charge current will be available to the battery for charging. If the battery is not already fully charged, it will be charged at a current rate equal to the maximum charge current minus the current drawn by the pass-through loads.

The principle concerns involved with pass-through charging are cycling (in the case of voltage-windowed charging) and degradation due to sustained float-voltage (in the case of float-voltage charging). Repeated cycling in windowed-charging is likely to be prevalent when the load is intermittent, due to charge termination and subsequent discharging of the battery by the load. Degradation due to forced float voltage is most significant when the battery is charged to full state-of-charge (4.2 volts/cell typically). This can be mitigated by charging the battery to a lower float voltage (i.e. 4.1 volts/cell).

CHAPTER 7

SOFTWARE SYSTEM

Base Station Software Management

The base station designed to provide contact charging consists of two main subsystems. The first is the power supply system, which manages and provides the output power to charge any docked UAV system. The second subsystem is the base station monitoring system. This is the system that monitors the internal and external conditions of the base station and the environment, and is used to track the efficiency and operating parameters of the base station. Each of these subsystems is run by a microcontroller, and has its own custom software running in real time.

The main controller for the base station is a Raspberry Pi 3 Model B+. This board contains a Broadcom BCM2837 System on a Chip (SoC), with 4 ARM Cortex-A53 CPUs, running at a total of 1.2GHz. The board has 1GB LPDDR2 (900 MHz) RAM, which is plenty to run the services for this system. The Raspberry Pi 3 also contains a 10/100 Ethernet Bridge, 2.4GHz 802.11n Wi-Fi card, and Bluetooth 4.1 Low Energy (LE) capabilities, in addition to several A/V outputs and interfaces. This embedded computer is running the latest version of Yocto Linux, a custom operating system for embedded devices, specializing in real time I/O. All the software and the operating system are stored on a 64GB microSD card plugged into the board. The Raspberry Pi is used to control the charging system, and the overall operation of the base station. It was chosen due to its vast amount of input/output support, including ethernet capabilities, as well as its open source software system, which makes it easy to configure for our specific needs. The Raspberry Pi maintains a constant internet connection, using the ASU Wi-Fi connection system. This

will be upgraded to a wires LAN connection when the base stations are officially installed in their designated locations.

A second, less powerful controller used in the base station is a standard Arduino Mega 2560 controller. This controller is used to gather all of the data from the internal and external sensors in the base station, for the monitoring sub system. This microcontroller is based on the ATmega2560 chipset, and provides 54 digital input/output pins, 16 analog inputs, and 4 UART (hardware serial ports). This board was chosen due to its speed and reliability in gathering sensor data, as well as its small energy footprint and ease of programming.

The Raspberry Pi microcontroller is continuously running one piece of software, which controls operation of the base station. This software is written in Python 2.7, and uses the following python libraries: NumPy (for mathematical data analysis), SciPy (for regression analysis), matplotlib (for plotting and pattern analysis), Requests (for HTTPS coding), and IPython (for general super-user work). The software constantly polls the AirBud API system, checking the status of the drone on the docking station. This is used to determine whether or not the drone (if present) needs to be charged, and if so to what battery levels. This information is provided by the fleet management system for the AirBud project. The raspberry Pi also constant polls all of the sensors it is connected to, in order to determine the state of the system. A simple limit switch connected to the contact rails notifies the system when a UAV lands on the base station. If the UAV needs to be charged, the raspberry Pi activates the power supply system, and allows the contact rails to receive output power from the CC/CV source. The raspberry Pi is also connected to several voltage sensors in the base station, and keeps a log of the voltage and current passing through the

system to UAV. If there are irregularities in these readings, or no voltage at all, an error message is sent to the AirBud server via an API call.

To allow for easier access to the live power data for the system, the Raspberry Pi has been configured to create its own WiFi network; using any WiFi-capable device, the real-time graphs and voltage/current data can be viewed in a webpage. The webpage is hosted at the address "voltage/." The only additional hardware required for this is an Edimax EW-7811Un wireless 802.11b/g/n USB Adapter, which has the ability to be used as an access point. A separate adapter was used for this in order to prevent any interference with the raspberry pi's internet connection (for the online API). Unfortunately, it does not have a default driver on the Pi for using it as an access point, so a custom Linux driver was separately installed and configured to allow for this operation.

Figure 14 is an image of the landing page, showing the real-time status of the supply voltages.

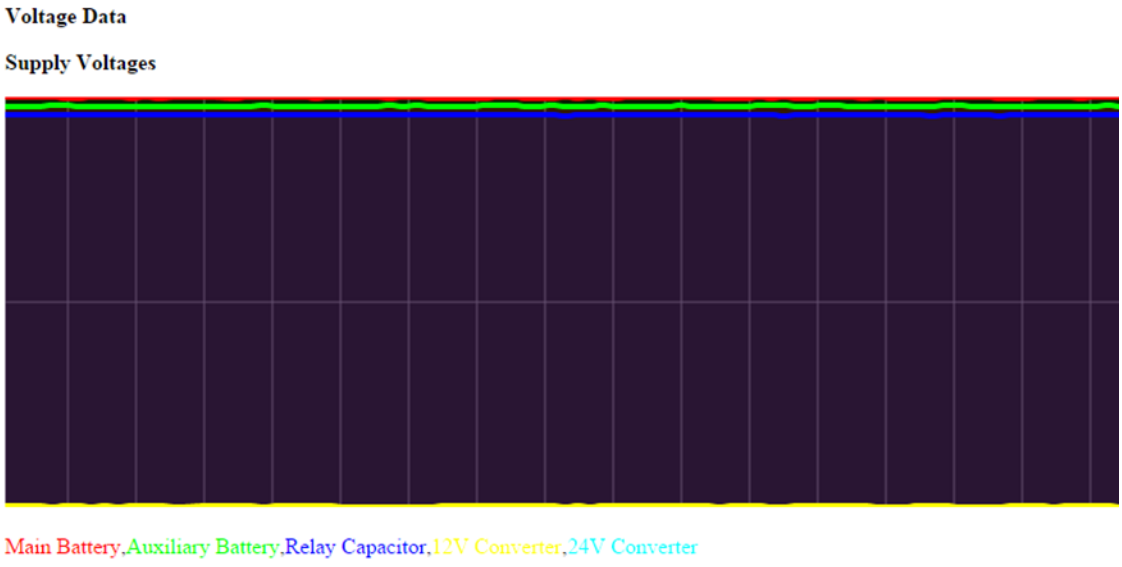


Figure 14: Landing page displaying real time voltage data in base station

The Arduino Mega microcontroller is connected to several sensors inside an outside the base station that gives it a real-time picture of the operating conditions of the pad at any given time. Several temperature sensors are connected using an Analog input pin, which are polled at a rate of 10Hz; These readings give a uniform temperature reading from inside the base station, to verify no devices are overheating. There is also a moisture sensor on the side of the station, which is used to detect any water leaks. If any moisture is detected inside the system, a signal is sent to the Raspberry Pi to immediately cut the power to the base station and all systems; this is to prevent any hardware damage or shorts due to water in the system. Outside the base station are a moisture sensor and a temperature sensor; these are both digital sensors that use the digital I/O pins on the Mega to relay information about the outside weather. This is used to notify the fleet management system if the base station needs to be covered or requires additional protection, in the case of harsh sunlight or heavy rain.

Onboard UAV Software Management

The UAV platform for this project currently uses an Intel Up Squared board. The Up Squared is based on a modern quad-core Intel Pentium chip, which allows it to run any x86 based operating system, and use any package compiled for x86. This consideration was made in order to allow the target environment to be the same as the development environment. As a result, the control software, the Intel RealSense cameras, and the 4G modem worked perfectly on the first attempt. The Up Squared board also includes a variety of input/output features. These include a mini PCIe port for the 4G modem, a large amount

of GPIO (general purpose input/output) pins for connecting the Pixhawk autopilot and 900 MHz modem, and USB ports and FTDI support for the camera modules.

Each UAV platform that works with this base station has its own onboard software that also monitors and manages the battery charging system. This is done to set the charge voltage levels on each battery, as they vary based on the drones' daily usage and schedule. This system is also used to monitor battery health on each UAV, in order to maximize battery lifetime and minimize maintenance. This software was custom created to work with the computer hardware and sensors aboard the UAV.

Each battery contains a BMS (battery management system) board which supports I2C and Serial communication protocols. The Up Board is wired to both of these outputs on the BMS, and maintains a constant connection throughout operation. The I2C connection is used to transmit all of the log data, while the serial connection is used to transfer error and BMS health data (in case the battery dies or a major component malfunctions). The Up Board runs a simple python script which polls the BMS board for battery state at a rate of 1Hz. The data provided by the BMS board includes the total charge of the battery in volts, the current estimated capacity, the current discharge rate, and the voltage value of each individual cell. This data is then processed and evaluated for anomalies, before being stored locally in a log file and transmitted to the fleet management system as part of a telemetry heartbeat. The current battery voltage levels and discharge rate are also transmitted to the Pixhawk module, which runs the autopilot; this data is used to optimize real time flight plans and warms the autopilot when the battery is near depletion. The individual cell voltage data is used to monitor battery health, and to verify a uniform discharge rate across the battery pack.

The autopilot software onboard the UAV systems is equipped with several safety features. One is a battery level limit function, which stops flight operations when the battery is nearly depleted, and returns the UAV to the nearest landing area. Another feature is the deployment of the parachute, which is triggered based on several factors, including battery level, UAV stability, and motor health. The autopilot and Up Board are connected to the TerminateAir module, which cuts off all power to the entire UAV system, allowing for safe parachute deployment. This module can be triggered using the software running on the Up board, such as in the case where the battery is shown to have reached a dangerous operation state or is completely depleted. This software is controlled by a watchdog system, outside of normal software operations.

UAV Battery Level Scheduling System

One of the main focuses of this project is to determine how to increase the longevity of a fleet of UAV systems during a long-term deployment. One of the biggest limiting factors for long-term missions is battery efficiency and health; as the rate of charging cycles and battery use increases, the battery lifetime and efficiency decreases. In addition, for robot systems that are inactive, storing a Lithium Polymer (LiPo) battery at full charge is also dangerous, and reduces the lifespan of the battery. In order to maximize battery lifespan and usage, the charge levels and usage of individual UAV battery packs needs to be optimized. Since LiPo battery packs cannot be stored at full charge, the best way to solve this problem is to choose some of the UAV systems to be at full charge, while leaving the rest at a 50% battery charge, which is the ideal level for LiPo storage. This way, there are enough UAVs at full capacity ready to meet all foreseen demands, while the rest of the

UAV platforms stay in a type of simulated "sleep mode," where they will almost never be used, hence increasing their battery lifespan. After a set amount of time, the roles of the UAVs in the fleet cycle, enabling the highly used vehicles to spend some time in "storage mode."

The main challenge is to determine just how many UAVs are needed to be kept at full capacity charge, and how many can be set at the optimal storage charge. This situation can be modeled as a constraint optimization problem. The goal of this optimization problem is to minimize the number of drones at full charge; this can be done by minimizing the total charge/capacity level of the entire fleet, using equation (1) with parameters (2) and variable ranges (3):

$$min_{c_i} \sum_{t=0}^{T} \sum_{i=1}^{N} c_i(t) \tag{1}$$

$$s.t. \sum_{i=1}^{N} \alpha_{c_i}(t) \geq E[D_t] \quad \forall t = 0, 1, 2, 3...T, \; c_i(t) \in \{v_1, v_2...v_m\} \tag{2}$$

$$\text{Charge values: } \{v_1, v_2...v_m\}, \text{ Capacities: } \{\alpha_{v_1}...\alpha_{v_m}\} \tag{3}$$

for N drones, T time steps, where c_i is the charge of the i-th drone at time step t, $E[D_t]$ is the expected demand at time t, α_{c_i} is the mission capacity of the i-th drone, and v_i is the voltage of the i-th drone. The charge values are a list of possible voltage charges for each UAV platform, which is preset in software. The capacities are the number of missions that each platform can accomplish based on the voltage state of the battery. A table showing the approximate values for these two variables in the Airbud UAV system is shown below. This optimization problem takes in discrete input, and is able to work at scale. The assumptions made for this implementation are that each UAV can serve the entire

operational area (full coverage), mission time is on average 15 minutes, mission capacity correlates linearity to battery capacity (each mission consumes the same amount of capacity), and the distribution of demand is uniform.

Table 4

UAV Battery States & Parameters

State	Percentage	Voltage (V)	Voltage per Cell (V)	Flight Time (min)	Charge Time (min)
Full Battery	100%	25.2	4.20	40	60
Military Spec	93%	23.4	4.00	35	45
Minimum Flight	55%	22.0	3.85	15-20	30
Storage	50%	21.6	3.80	10	30
Dead Battery	0%	18.0	3.00	0	N/A

There are five different states defined in this table. Full battery is when the battery it at 100% capacity, it cannot hold any more power; the UAVs will never have their batteries at this level as it reduces lifespan. Military Specification is the industrial standard for how high a LiPo battery should be charged. The UAV systems that will be at "full charge" will be charged to this level. The minimum flight time state is the lowest a battery level can be where the UAV will be allowed to accept a mission; any UAV with a battery level lower than 55% will not be able for dispatch to a mission request. This is a safety feature that has been set in software in the fleet management system. Storage is the state the LiPo batteries will be in when the UAVs are not in use. This equates to approximately 50% battery capacity. At this level, a LiPo battery can be stored for weeks without any

internal damage or issues; the trickle discharge rate is also minimum. Each UAV that is not chosen to be at full charge will be stored at this state. It will take the base station five to ten minutes to prepare a UAV for flight readiness from the storage state, and 15 minutes to charge the battery up to military specifications (23.4 volts). The last state is the dead battery state, which is reached when the LiPo battery is at a nominal 18-volt charge. The UAV LiPo batteries should never reach this state, except in standard maintenance discharge cycles. This table also has the approximate flight time for each UAV based on battery level, as well as the charge time (from the dead battery state).

For implementation of this optimization algorithm, a python script was created, using the Pyomo optimization library. The program has several preset parameters, including the voltages in the table above, the average mission flight time and power consumption, and the number of time steps (T) per day. The inputs for the program are the number of total drones in the system, and the expected demand for the current time step. The expected demand is calculated using the historical data for UAV flight missions over the past 14 days, as well as other factors. This script runs in the GCE cloud server, alongside the fleet management system, and determines how many UAVs should be set at each charge rate; currently the only two charge rates are Military Spec and Storage. This data is then given to the fleet management system, which assigns charging states to individual UAVs and their respective base stations, based on location and demand. This script is called to run once every hour, since the time steps used in the equation are in 60 minutes apiece. A UAV platform can be moved from the storage state to the charged state after one-time step (in order to meet demand), but the program is forced to wait for four times steps

before it will allow a UAV platform to move from charged to storage, just in case demand increases again.

CHAPTER 8

TESTING & IMPLEMENTATION RESULTS

Desktop Testing

In order to determine the ideal system components and settings for this system, a large amount of simulation and desktop testing was completed prior to full assembly. Each of the hardware subsystems was tested using prototype parts and production ready parts, and every piece of data was logged. A full testing plan and data log is available in the appendix.

Before any circuit construction was completed, the entire power generation system was designed in a circuit designer software, EagleCAD. These circuit designs, pictured below, were then imported into PSPICE circuit simulation software. Using this software, a variety of different input signals and loads were tested on the circuit systems, to verify the outputs would meet the proper requirements without fail. This simulation testing also allowed for stress testing of the system, in order to determine the operating constraints of the circuits before system instability; this led to changes in several of the parts to industrial-grade components (most notably the main PSU unit and Schottky Diodes) to maintain system operation for all possible use cases.

After the circuitry was chosen and purchased, a simple desktop system was setup in order to test all of the circuit components together. For this hardware in the loop testing setup, the power supply unit was connected directly to the DC-DC buck converter, which attached to the off-the-shelf drone battery BMS system. A voltage/current monitor board was also connected in series in order to pull log data from the system. The results of this test were to show that the power output of the DC-DC converter was able to match the

requirements of the battery BMS system, without going over. A nominal output voltage of 23.4 volts was shown consistently, at a rate of 12 Amps, which is the expected output for this system.

In order to determine the best material to use for the drone and base station contacts, several different metals were tested using the desktop setup. In this case, two separate pieces of the metal were connected to the circuitry: One connected to the DC-DC buck converter, and the other to the battery management board on the drone battery. The metals tested were: Brass, Aluminum, Gold-plated copper, nickel-plated copper, graphite (brushes), and pure copper. A diagram of the setup used for the test (for each metal) is shown in Figure 15.

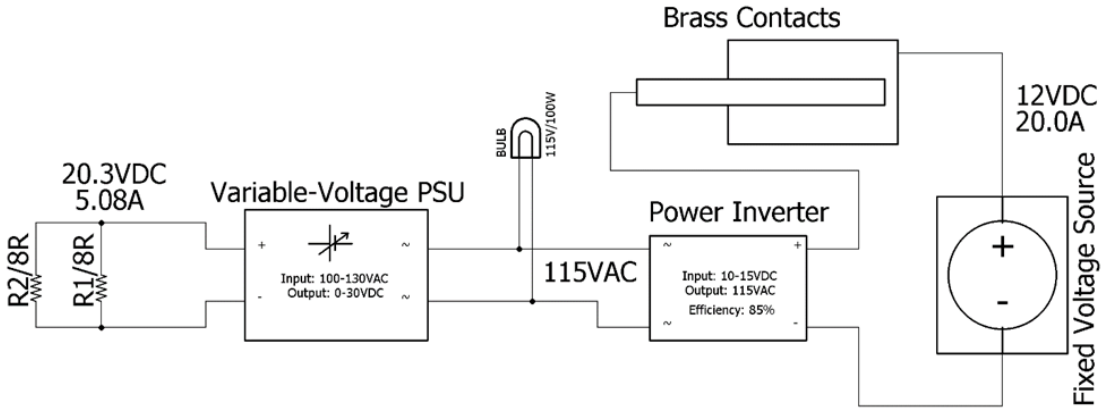


Figure 15: Schematic of testing setup used for brass conductor tests.

After an initial analysis, it was decided that the best conductors were the copper and brass. Ultimately, the brass was chosen due to its greater resistance than copper, which is ideal for long term use. Brass sheets were used as the contact points in the desktop setup, to create the testing setup shown below. The goal of this test was to record the contact resistance, voltage drop, and temperature readings of the brass, to ensure its durability and affinity for the system. Table 5 shows the results.

Table 5

Brass Conductor Capabilities Testing Results (Current, Temperature, & Resistance)

Time Elapsed (Min):	Current (A):	Ambient T (C):	Peak Brass T (C):	Contact Resistance (Ohms):
0	20.45	23.3	23.3	0.003471883
5	20.42	24	24.2	0.002757101
10	20.36	24.4	24.8	0.002799607
15	20.61	24.7	26.2	0.002993692
20	20.23	24.4	26	0.002970835
25	20.36	25.1	26.7	0.002892927
30	20.6	25.5	27	0.003106796
35	20.6	25.4	27	0.002635922
40	20.54	24.8	26.3	0.002322298
45	20.35	25.7	27	0.002452088
50	20.61	25.7	27.2	0.002717128
55	20.61	25.4	26.8	0.002911208
60	20.16	25.9	27.1	0.003323413

Another desktop test that was completed was testing the control system of the base station, powered by the raspberry pi microcontroller. The microcontroller was connected to the 250V/10A logic-level relay, used to start and stop the power supply unit, as well as the limit switch in the base station, which signals when the UAV platform is connected to the base station. When the limit switch was activated, it sent a signal to the raspberry pi, notifying it of the UAV's presence. The Pi would then check the fleet management system, to verify the UAV's need for charging. If charging was approved, the relay switch activated using a standard digital HIGH output from the Pi, which would allow power from the PSU to power the rest of the charging system. The relay immediately switches off as soon as the limit switch is depressed, or when the Pi is told the drone no longer needs to be charged. This system was tested 25 times, with all possible inputs and configurations. This system

was also tested with the data from the moisture sensor, which disables the entire system whenever moisture is detected in the system.

The last desktop test was a discharge test for the power storage systems. This was performed on the batteries used for the UAV system, both the COTS battery and the custom LiPo battery built from scratch. For this test, a variable load was connected to the battery through both direct wires and the brass contact connections. The variable load consisted of two 8 Ohm resistors, each capable of handling a 100 Watt load, strapped to a large heat sink. These resistors were connected first in parallel (for a 4 Ohm load) and then in series (for a 16 Ohm load) for testing with the battery. The current and voltage in the system was constantly monitored, and compared to the battery's factory specifications to verify no abnormalities. The main test was with the 4 Ohm load, since this is similar to the load of the drone during operating systems. For the 4 Ohm load test, the battery was connected while at full capacity (25.2 Volts), and was run until the battery was fully empty (18 volts). The current through the system started at 6.3 Amps, and gradually dropped to 4.5A, until the battery was finally drained. The battery management system broadcasted a visual and audio warning when the battery was at 15% capacity, and automatically shut off the battery once it hit 18 volts, which is essentially an empty battery (any lower is dangerous). On average, the discharge time of the battery was approximately 45 minutes, at a 1C discharge rate. This is in line with our standard operating procedures. This same test was repeated with both 8 Ohm and 16 Ohm loads, with similar results.

Full System Testing

There are three main tests that were completed for the full system testing stage. The testing plan is provided for the last test, as an example.

The first test was of the entire base station hardware (open access) and the power system of the UAV platform (detached from any other systems). This did not include the pass-through system designed for the UAV power system. The UAV power system was connected directly to the contact points on the base station, which was set to charge the battery at a rate of 1C, up to 25 volts (near max capacity). The current was monitored throughout this process, and maintained a constant 12 Amps (the 1C rate) until shutoff. There were no detected anomalies or spikes in power output, and the battery was fully charged in 40 minutes.

The second test consists of the completed (sealed) base station and the final version of the UAV platform, including all onboard circuity and control software. This included the pass-through system designed for the UAV. The UAV was placed on top of the base station, in standard landing configuration, and the base station was sent the command from the fleet management system to begin charging. Battery level and charging rate were monitored both on the UAV and the base station throughout the test; battery level increased at the expected rate and the charge rate was uniform at 12 amps. The battery charged up to 23.4 volts and then the BMS board switched to a trickle charge, while giving the pass-through system priority.

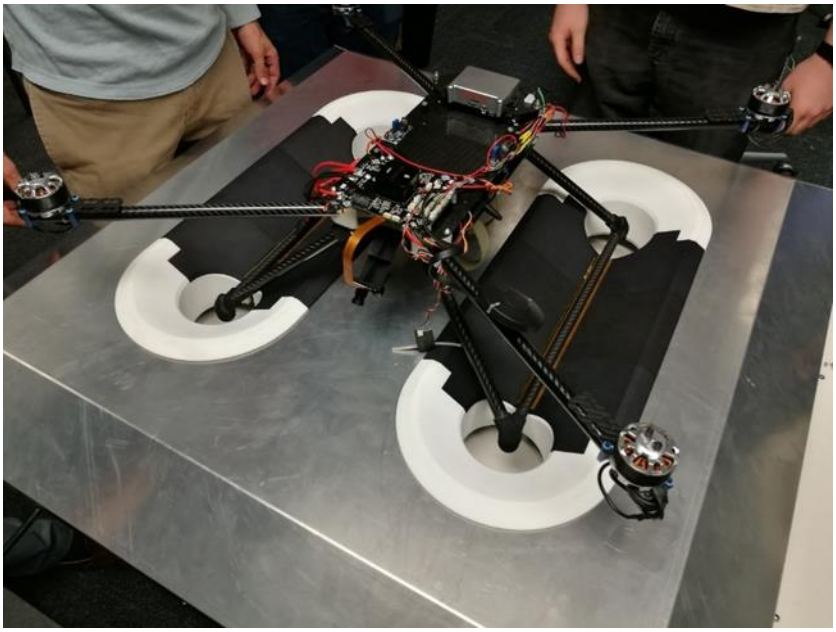


Figure 16: Testing setup for full system integration tests

There was also a separate test for the passthrough system of the UAV power subsystem, due to its major importance to this project as a whole. This test was where the UAV pulled power directly from the base station for normal operations, instead of just charging the battery. The testing plan used to validate this system is in the next section.

General Pass-Through Testing

Pass-through charging performance can be characterized under the following conditions:

- No pass-through load (standard charging)
- Pass-through load less than 50% of charging rate
- Pass-through load greater than 50% charging rate and less than 100% charging rate
- Pass-through load greater than 100% charging rate (augmented discharging)

Under each condition, the battery should be charged from 0% state-of-charge to 100% state-of-charge with the operating pass-through loads connected, except greater than 100%

charging rate. When state-of-charge reaches 100%, the following steps should be followed for all except greater than 100% charging rate:

- Choose a charge termination profile (voltage-windowed, float-voltage, etc.)
- Operate the battery in the desired charging profile for a pre-defined amount of time (i.e., 24 hours, 48 hours, etc.)
    - For voltage-windowed charging, determine the number of window-cycles that occur at each load level after the battery reaches 100% state-of-charge.
    - For float-voltage charging, remove the pass-through load at the start and conclusion of the 100% state-of-charge testing period and measure the current flowing into the battery. Determine if any notable changes in quiescent battery current occur.
- For greater than 100% charging rate:
- Connect the pass-through load and initiate charging.
- Monitor the battery voltage and current during operation.
- Manually interrupt the load when the battery has reached 0% state-of-charge.

Noise and Stability Pass-Through Testing

The following loads should be characterized during the testing procedure (same as above):

- No pass-through load (standard charging)
- Pass-through load less than 50% of charging rate
- Pass-through load greater than 50% charging rate and less than 100% charging rate
- Pass-through load greater than 100% charging rate (augmented discharging)

In addition to DC voltage and current measurements at the battery terminals, an oscilloscope should be connected to the battery terminals and set to AC-coupling mode. The test should be started with battery state-of-charge moderately above 0% (5%-10%) in order to prevent over-discharging during the initial non-charging load tests. At each of the following charging conditions, the RMS and MAX AC voltages present on the battery line should be recorded:

- No load, no charging
- Test load, no charging
- Test load, charging, 5-10% SoC
- Test load, charging, 25% SoC
- Test load, charging, 50% SoC
- Test load, charging, 75% SoC
- Test load, charging, 100% SoC (initial)
- Test load, charging, 100% SoC (over a 24-48 hour period)

The RMS and MAX AC voltages for each of these conditions will be recorded and any significant aberrations will be noted as potential issues. Possible mitigation techniques will include introduction of electrolytic capacitors and low-ESR ceramic capacitors in parallel to the battery.

CHAPTER 9

INDUSTRY CERTIFICATION & APPROVAL

The eventual goal of this project is to operate both the base stations and UAV systems in an urban environment. In addition to all of the technical challenges that have to be solved, there is a myriad of legal requirements that must be satisfied. The Federal Aviation Administration (FAA) is the governing body in charge of regulating all civilian aviation in the United States, which now includes unmanned aerial vehicles (UAVs). Any aircraft or aerial vehicle that plans to be flown in the national airspace (NAS) must receive approval from the FAA, either in the form of a standard aircraft certificate, or an experimental certificate. A thorough analysis of current FAA regulations and possible solutions can be found in Appendix B; this report includes a case study of the Arizona State University (ASU) Tempe location. Although currently research and testing for this project is being completed using a Part 107 certification (hobbyist license), in the future this system will need to be certified either through the FAA's experimental aircraft program, or through an OEM manufacturer. The requirements for FAA experimental certificates is as follows:

> 21.195 Experimental certificates: Aircraft to be used for market surveys, sales demonstrations, and customer crew training. (a) A manufacturer of aircraft manufactured within the United States may apply for an experimental certificate for an aircraft that is to be used for market surveys, sales demonstrations, or customer crew training. (b) A manufacturer of aircraft engines who has altered a type certificated aircraft by installing different engines, manufactured by him within the United States, may apply for an experimental certificate for that aircraft to be used for market surveys, sales demonstrations, or customer crew training, if the basic aircraft, before alteration, was type certificated in the normal, acrobatic, commuter, or transport category. (c) A person who has altered the design of a type certificated aircraft may apply for an experimental certificate for the altered aircraft to be used for market surveys, sales demonstrations, or customer crew training if the basic aircraft, before alteration, was type certificated in the normal, utility, acrobatic, or transport category. (d) An applicant for an experimental certificate under this section is entitled to that certificate if, in addition to meeting the requirements of § 21.193— (1) He has established an inspection and maintenance program for the continued airworthiness of the aircraft; and (2) He shows that the aircraft has been flown for at least 50 hours, or for at least 5 hours if it is a type certificated aircraft which has been modified (Amdt 21-70, 1992).

In order to apply for the experimental certificate, this system needs to meet three criteria: an inspection and maintenance program for continued validation, and flight testing of an excess of 50 hours, or 5 hours if it's a modified aircraft. Since these UAVs are custom built platforms, flight testing for greater than 50 hours is required. This goal will be met by the end of April 2018. As for safety requirements, aircraft worthiness, and inspections and maintenance protocols, we have decided to emulate the standards and procedures used by Dà-Jiāng Innovations, better known at DJI. DJI is a Chinese based drone company, and the main industry leader in terms of commercial UAV and drone systems. They have received a Section 333 exemption from the FAA for their drone platforms, as well as several experimental certifications for their products. The DJI Inspire platform is closely related to the custom UAV platform built for Project Airbud, so we are using the Inspire protocols for safety and maintenance. The UAV power system has been built to meet all of DJI's power standards, which can be found in Appendix C, under the ESC standards. A

representative from Gresco, DJI's main distributor in the southwest, has personally inspected the UAV battery system in use for this project, and has declared it exceeds DJI's battery standards. This inspection included long-term endurance and durability testing of the COTS battery, as well as a side-by-side breakdown comparison of a standard DJI battery and this battery.

Lastly, in order to assure that all flight plans and operations will be conducted safely and legally in the ASU airspace, ASU and the Project Airbud team has started working with the local government on UAV policy efforts. A major part of this is the state of Arizona's bid for the UAS Pilot Program, a detailed description of which can be found in Appendix B, page 86. Part of the work put forth by the UAS committee has been flight maps and boundaries for all UAV operations in the greater Phoenix metro area. We helped shape and map out these flight areas, and have included the entire ASU Tempe campus in the allowable area. We have also created software that constantly checks existing regulations and verifies flight plan compliance before mission deployments (see Appendix B for documentation). Another focus of this group has been to put together a list of requirements for airworthiness for all AUV platforms operating in this airspace. Currently, these requirements mirror those of DJI and 3D Robotics, another large UAV company. Since these standards are based off the same as used in this project, Project Airbud and this research project's system both meet these standards and are cleared for operation in the future.

CHAPTER 10

KEY CONTRIBUTIONS & OUTCOMES

The outcome of this project was the successful design, implementation, and deployment of a contact charging base station for use with a fleet of unmanned aerial vehicle (UAV) platforms. Using engineering design principles, this landing pad was built from the ground up, optimized for manufacturing and easy scalability. A robust power management system was developed both for the charging pad and for the on-board UAV battery system. A complex power-optimization algorithm was also developed for use by a UAV fleet management system, to regulate the number of vehicles that need to be fully charged at any given time; this allows for extended battery lifespans for the fleet as a whole. A fully-functioning prototype of this base station was designed and tested, and was built in parallel with a custom UAV platform with contact charging capabilities. The base station is able to be installed in any location, provided there is a standard AC power source and some form of internet connection available. It is able to monitor its surrounding environment and react to any potential issues during normal operations, such as rainfall or battery failure. The UAV system uses a custom battery that is designed to work in tandem with the contact charging system, and has been hardened to work in extreme deployment environments. Through several sets of testing and deployments, it was shown that this complete system allows for extended autonomous operations in a confined area, and improves over existing solutions to power consumption/resource problems in autonomous robotics.

CHAPTER 11

FUTURE WORK

There are several areas in which this project and research can be improved in major ways. The first is the battery charge level optimization algorithm. The equations designed for this project are limited in their scope, as they were designed with the Project Airbud system in focus. The algorithm does not consider varied mission times, as it assumes that each mission takes the same timespan (and hence battery capacity) on average. This optimization does not include a spatial component as well; it does not take into account platform locations relative to demand areas in the operating environment. For this project, we worked under the assumptions that each unmanned aerial vehicle (UAV) could perform a mission in any location in the operating area, and mission demand is uniform across the area, so location-based optimization is not necessary in mission scheduling. For a scalable implementation of this, the algorithm needs to be updated to include a spatial component, as well as adaptation for changes in demand over time in different locations. The fleet management system that uses this algorithm should also be upgraded, with the capabilities to monitor and predict demand for different parts of the system. This can be done by analyzing the number of mission requests that are received on a per-day basis, and constructing a simple model to predict demand behavior throughout the week. This would be extremely beneficial for the system, especially if demand is clustered in several prominent areas, as it would allow for better optimization of UAV platform locations as well as drone readiness and charge levels.

The base station design also has several planned improvements. The first is to improve the waterproofing/weatherproofing of the shell. This will ideally be done through

further anodization, as well as creating a waterproof seal between the shell layers using a rubber insolation layer and potentially waterproof electrical connectors (subconns). Another update is to develop a way for the base station to establish a data-transfer connection with the UAV using the landing pad. Currently, both the base station and the UAV have separate internet connections, which they use to communicate with the cloud API server which collects all of their data and manages the fleet. Ideally, the UAV should connect to the base station and relay information directly to the charging control, requesting power and other input only when needed. As a bonus, the base station can be used as a data dump for UAV flight data and telemetry logs post-mission, allowing for a faster transfer rate and less down time. The system in the base station can also then be used to ascertain a better estimate of UAV health status, for both the battery and general propulsion systems, through a direct connection and test protocol. A last suggested upgrade for the base station would be to increase its capacity, allowing for multiple UAV platforms to use a base station at a given time. This could be accomplished by connecting several base stations together, or modifying the current design to accommodate several systems at a given time. This would allow for a higher density of UAV platforms to be based in a set operating area.

The last area of improvements that can be implemented is on the UAV power system side. Currently, an off the shelf battery pack with a built-in BMS is being used on the UAV system. Although we have created a custom battery pack for the UAV, it is not as compact or durable as the COTS model. Future work can focus on building a better custom battery for use with this charging system, focusing on higher heat tolerances for LiPo cells and custom firmware for the BMS board. In particular, allowing the onboard autopilot to control settings on the BMS will be an important milestone because it will

allow for better battery health diagnostics as well as more effective battery charging and discharging.

CHAPTER 12

CONCLUSION

The main goal of this thesis project was to develop a new contact charging solution for unmanned aerial vehicles (UAVs) to use during long-term deployments. Specifically, the focus was to create a system that would be used on the ASU Tempe campus in tandem with the new UAV safety system project run by the Office of Knowledge Enterprise and Development. During the course of this project, a base station platform was designed to deliver a variable power source through a contact-based physical connection, to a custom battery system designed for a test UAV. The system was first designed using computer aided drafting (CAD) and simulation software, and eventually was fabricated using of-the-shelf components. A custom battery and power system were designed for UAV platforms that would work with this contact charging system; it was implemented on a test vehicle, which was then used to rigorously test the base station prototype. The results were that the base station supplies a constant, configurable power source that is more than sufficient to charge any type of standard lithium polymer battery. In addition to the hardware that was created for this project, custom software was written and implemented that controls not only the charging system for each base station, but the entire fleet-wide charging system. This software includes an adaptive optimization algorithm that allows the system to determine the most efficient use of each UAV platform, to optimize fleet operations over a large period of time. The software was successfully integrated into the existing fleet management system, and is currently in use as a part of the UAV management system at ASU. This project is continuing to undergo extreme testing, with integration plans for the ASU Tempe campus currently under review. Overall, the attempt to design an efficient,

adaptive power management system for UAV fleets was successful, and the prototypes are currently being refined and prepped for manufacturing at scale.

REFERENCES

Al Juheshi, F. et al. (2017). Novel adaptive battery system for integration with multi-level inverters, Electro Information Technology (EIT) 2017 IEEE International Conference on, (pp. 066-070). IEEE

Amdt. 21-70, 57 FR 41369. (1992).

Baronti, F., Fantechi, G., Leonardi, E., Roncella, R., & Saletti, R. (2011, September). Hierarchical platform for monitoring, managing and charge balancing of LiPo batteries. In Vehicle Power and Propulsion Conference (VPPC), 2011 IEEE (pp. 1-6). IEEE.

Bellingham, J. G., & Rajan, K. (2007). Robotics in remote and hostile environments. Science, 318(5853), 1098-1102.

Bethke, B., How, J., & Vian, J. (2009, August). Multi-UAV Persistent Surveillance with Communication Constraints and Health Mangement. In AIAA Guidance, Navigation, and Control Conference (p. 5654).

Borroni-Bird, C. (2013, August). Enabling connected and electric vehicles. In CAR management briefing seminars, Traverse City, Michigan (pp. 5-8).

Buchmann, I. (2018). Batteries in a portable world: a handbook on rechargeable batteries for non-engineers. Ec & M Books. Retrieved from http://batteryuniversity.com/learn/article/nickel_based_batteries

Campbell, J., Pillai, P., & Goldstein, S. C. (2005, August). The robot is the tether: active, adaptive power routing modular robots with unary inter-robot connectors. In Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on (pp. 4108-4115). IEEE.

Divakar, B. P., Cheng, K. W. E., Wu, H. J., Xu, J., Ma, H. B., Ting, W., ... & Leung, C. H. (2009, May). Battery management system and control strategy for hybrid and electric vehicle. In Power Electronics Systems and Applications, 2009. PESA 2009. 3rd International Conference on (pp. 1-6). IEEE.

Dunn, Terry (March 2015). Battery Guide: The Basics of Lithium-Polymer Batteries. Tested. Whalerock Industries.

Fujii, K., Higuchi, K., & Rekimoto, J. (2013, December). Endless flyer: a continuous flying drone with automatic battery replacement. In Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)(pp. 216-223). IEEE.

Goldstein, S. C., & Mowry, T. C. (2004). Claytronics: A scalable basis for future robots.

Indoor Charging Pad Technical Specifications. (2018). Skysense - Drone Charging Pad. Retrieved from http://www.skysense.co/charging-pad-indoor/specs

Kang, K., Meng, Y. S., Bréger, J., Grey, C. P., & Ceder, G. (2006). Electrodes with high power and high capacity for rechargeable lithium batteries. Science, 311(5763), 977-980.

Kit, D. et al. (2018). Wireless power | WiBotic. WiBotic. Retrieved from https://www.wibotic.com/wireless-power/

Lee, Y. S., & Cheng, M. W. (2005). Intelligent control battery equalization for series connected lithium-ion battery strings. IEEE Transactions on Industrial electronics, 52(5), 1297-1307.

Li, S., & Zhang, C. (2009, March). Study on battery management system and lithium-ion battery. In Computer and Automation Engineering, 2009. ICCAE'09. International Conference on (pp. 218-222). IEEE.

Manenti, A., Abba, A., Merati, A., Savaresi, S. M., & Geraci, A. (2011). A new BMS architecture based on cell redundancy. IEEE Transactions on Industrial Electronics, 58(9), 4314-4322.

Mian, S. (2016). Design and Implementation of an Electronic Preventative Maintenance System for Autonomous Vehicles. Repository.asu.edu. Retrieved from https://repository.asu.edu/items/37849

Nishi, Y. (2001). Lithium ion secondary batteries; past 10 years and the future. Journal of Power Sources, 100(1-2), 101-106.

Qiang, J., Yang, L., Ao, G., & Zhong, H. (2006, December). Battery management system for electric vehicle application. In Vehicular Electronics and Safety, 2006. ICVES 2006. IEEE International Conference on (pp. 134-138). IEEE.

Redding, J. D., Toksoz, T., Ure, N. K., Geramifard, A., How, J. P., Vavrina, M., & Vian, J. (2011, August). Persistent distributed multi-agent missions with automated battery management. In AIAA Guidance, Navigation, and Control Conference (GNC).

Schneider, B. (2018). A Guide to Understanding LiPo Batteries. Roger's Hobby Center. Retrieved from https://rogershobbycenter.com/lipoguide/

Scrosati, B., Abraham, K. M., van Schalkwijk, W. A., & Hassoun, J. (Eds.). (2013). Lithium batteries: advanced technologies and applications (Vol. 58). John Wiley & Sons.

Stuart, T. A., & Zhu, W. (2011). Modularized battery management for large lithium ion cells. Journal of Power Sources, 196(1), 458-464.

Sundaram, S. M., Kulkarni, M., & Diwakar, V. (2015, August). Management of large format liion batteries. In Transportation Electrification Conference (ITEC), 2015 IEEE International (pp. 1-7). IEEE.

Toksoz, T. (2012). Design and implementation of an automated battery management platform (Doctoral dissertation, Massachusetts Institute of Technology).

Ure, N. K., Chowdhary, G., Toksoz, T., How, J. P., Vavrina, M. A., & Vian, J. (2015). An automated battery management system to enable persistent missions with multiple aerial vehicles. IEEE/ASME Transactions on Mechatronics, 20(1), 275-286.

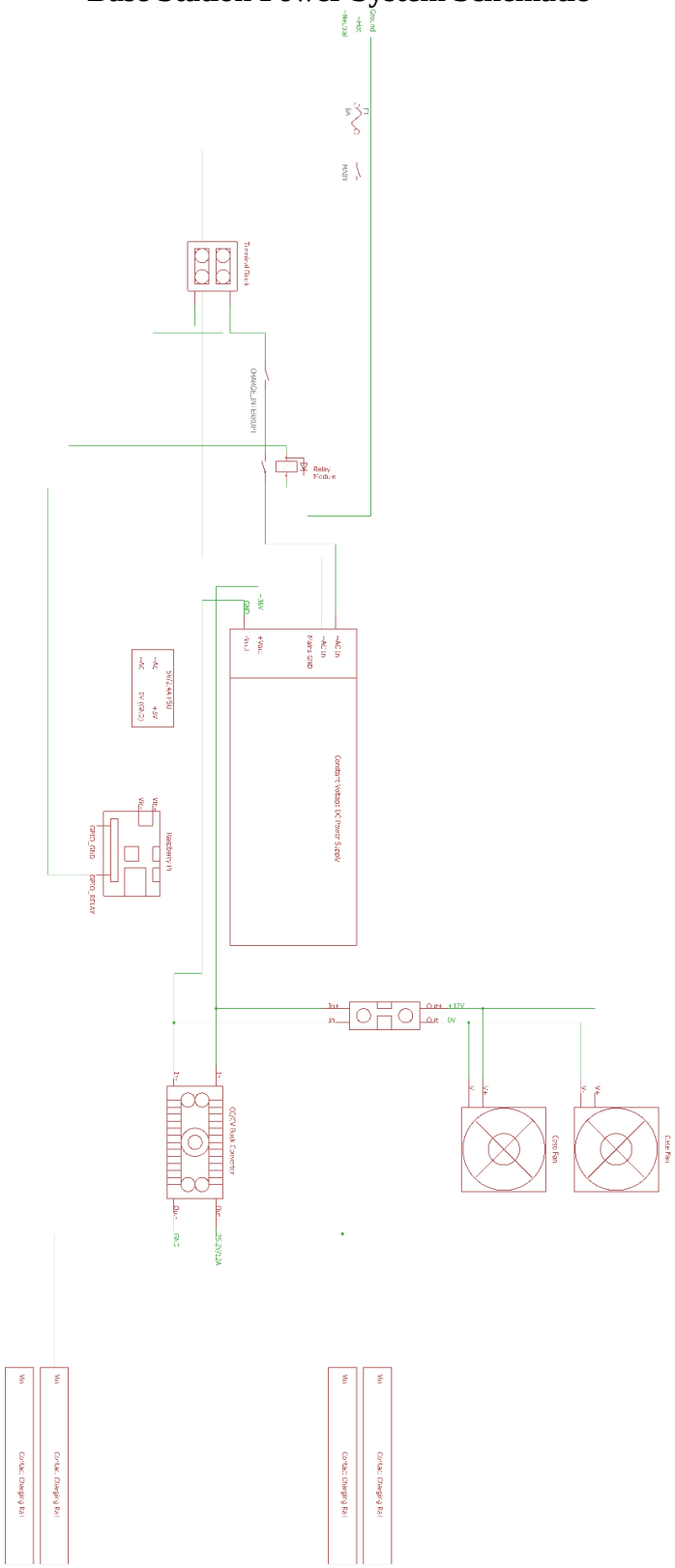
Voth, D. (2002). Nature's guide to robot design. IEEE Intelligent Systems, 17(6), 4-6.

Zhou, F., Lin, H., Hu, J., Lv, Z., Qian, B., & Xu, J. (2014, November). A novel balancing strategy for series-connected lithium batteries based on mixed charging mode. In Power Electronics and Application Conference and Exposition (PEAC), 2014 International (pp. 732-736). IEEE.

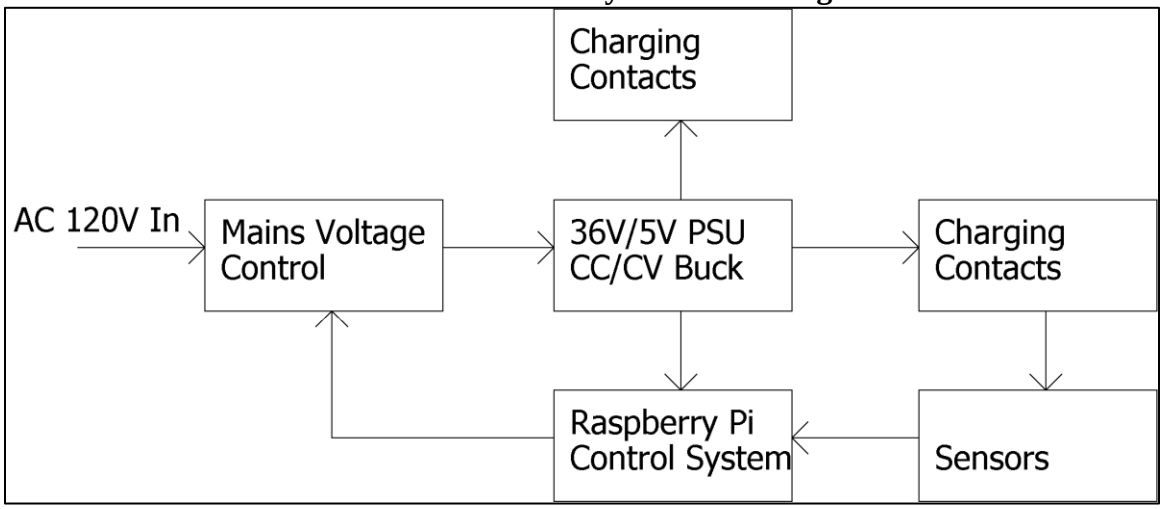
APPENDIX A

ELECTRICAL SCHEMATICS

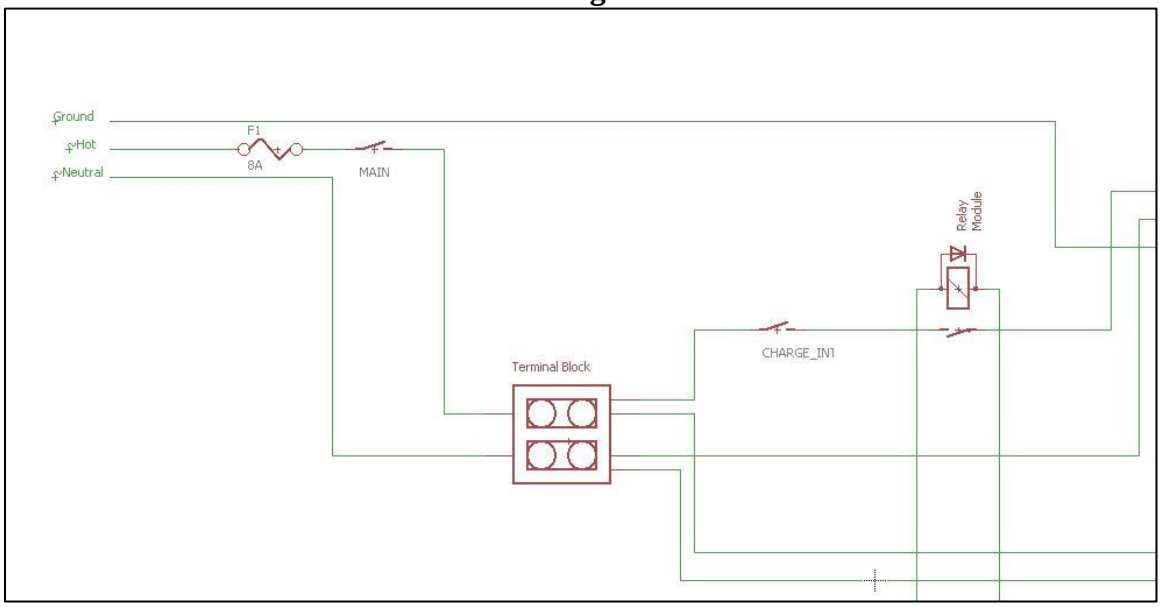
Base Station Power System Schematic



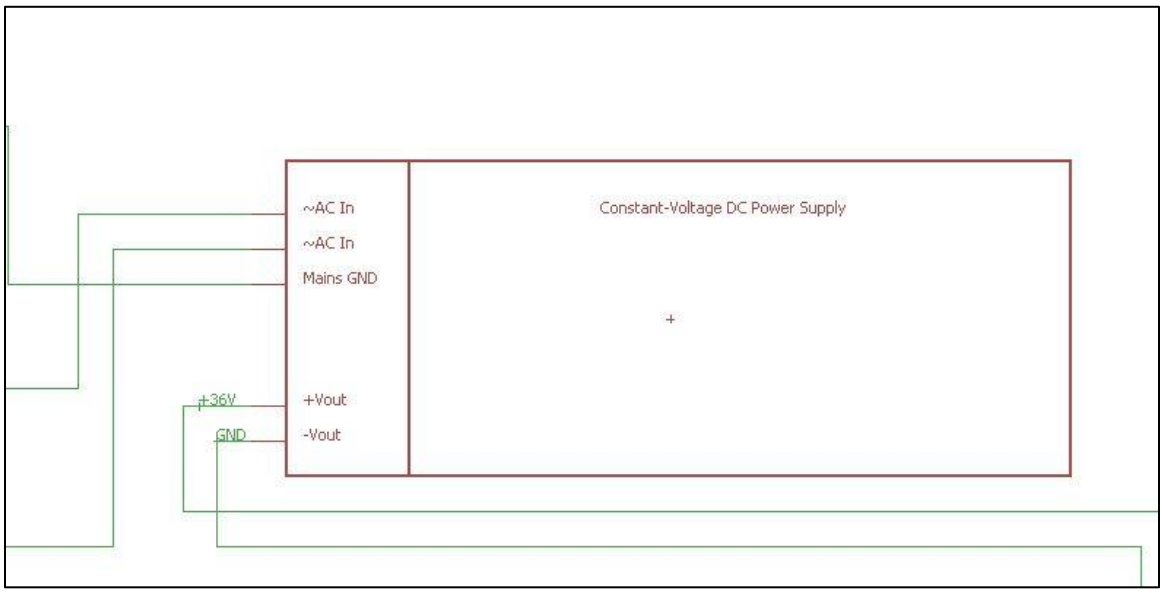
Base Station Power Subsystem Flow Diagram



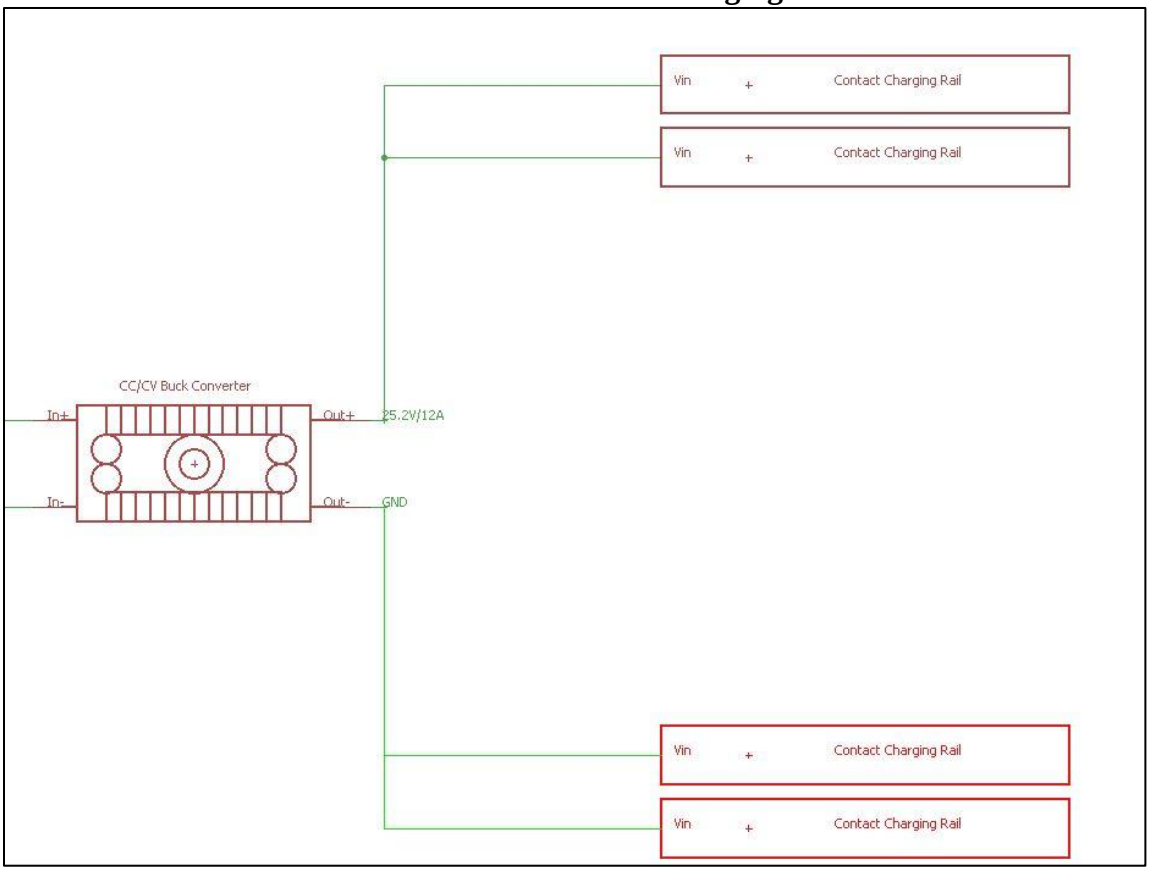
Mains Voltage Control



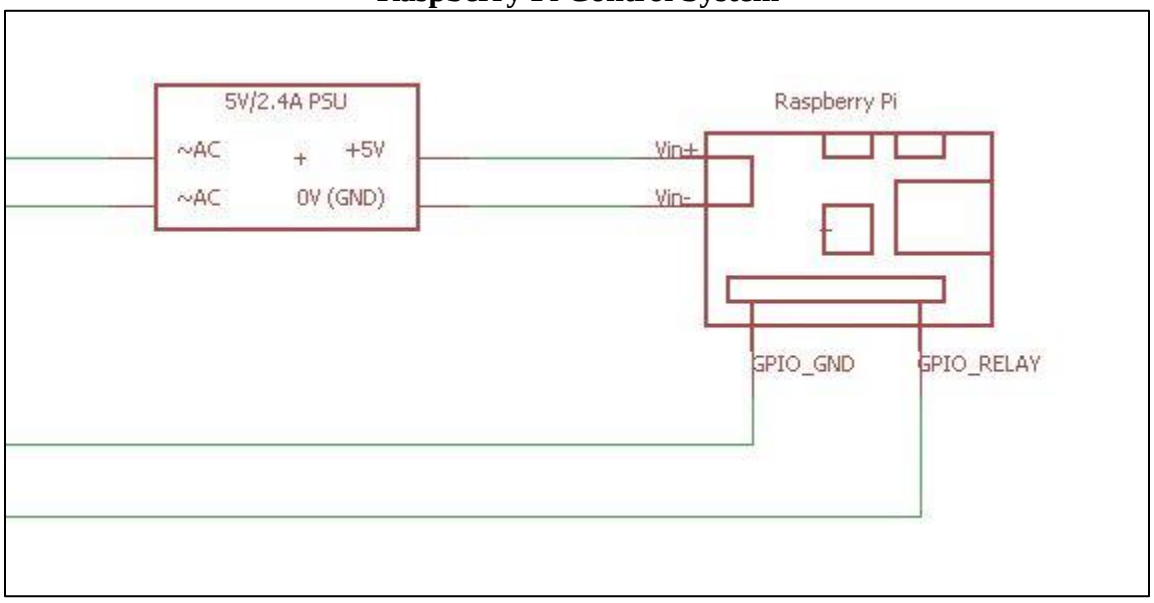
36V/5V PSU



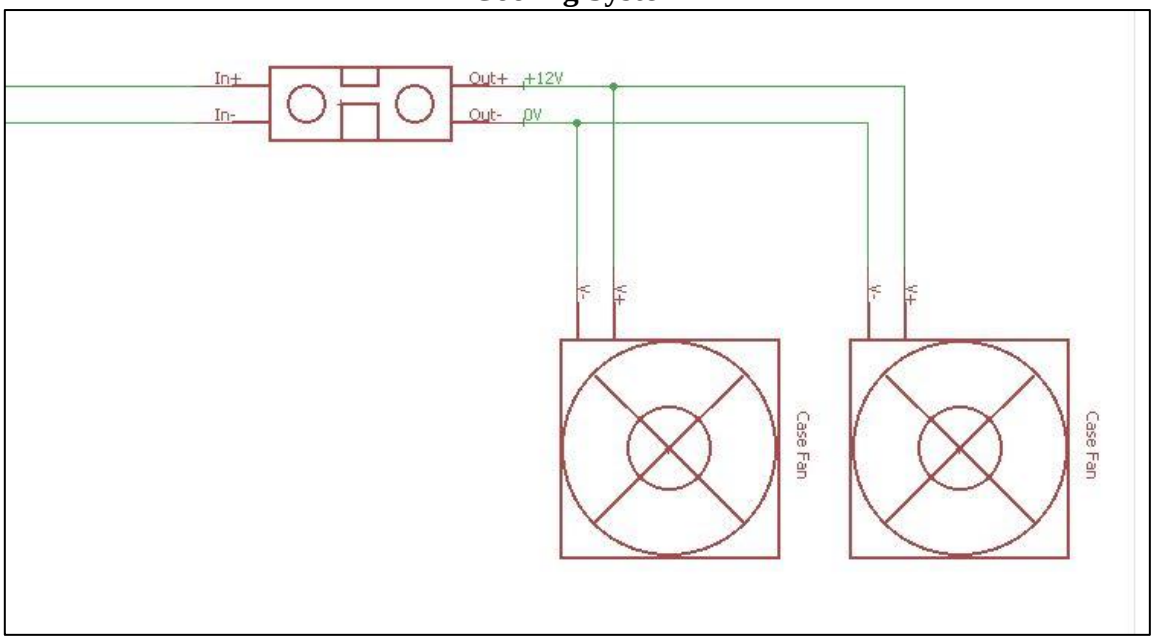
CC/CV Buck and Contact Charging Rails



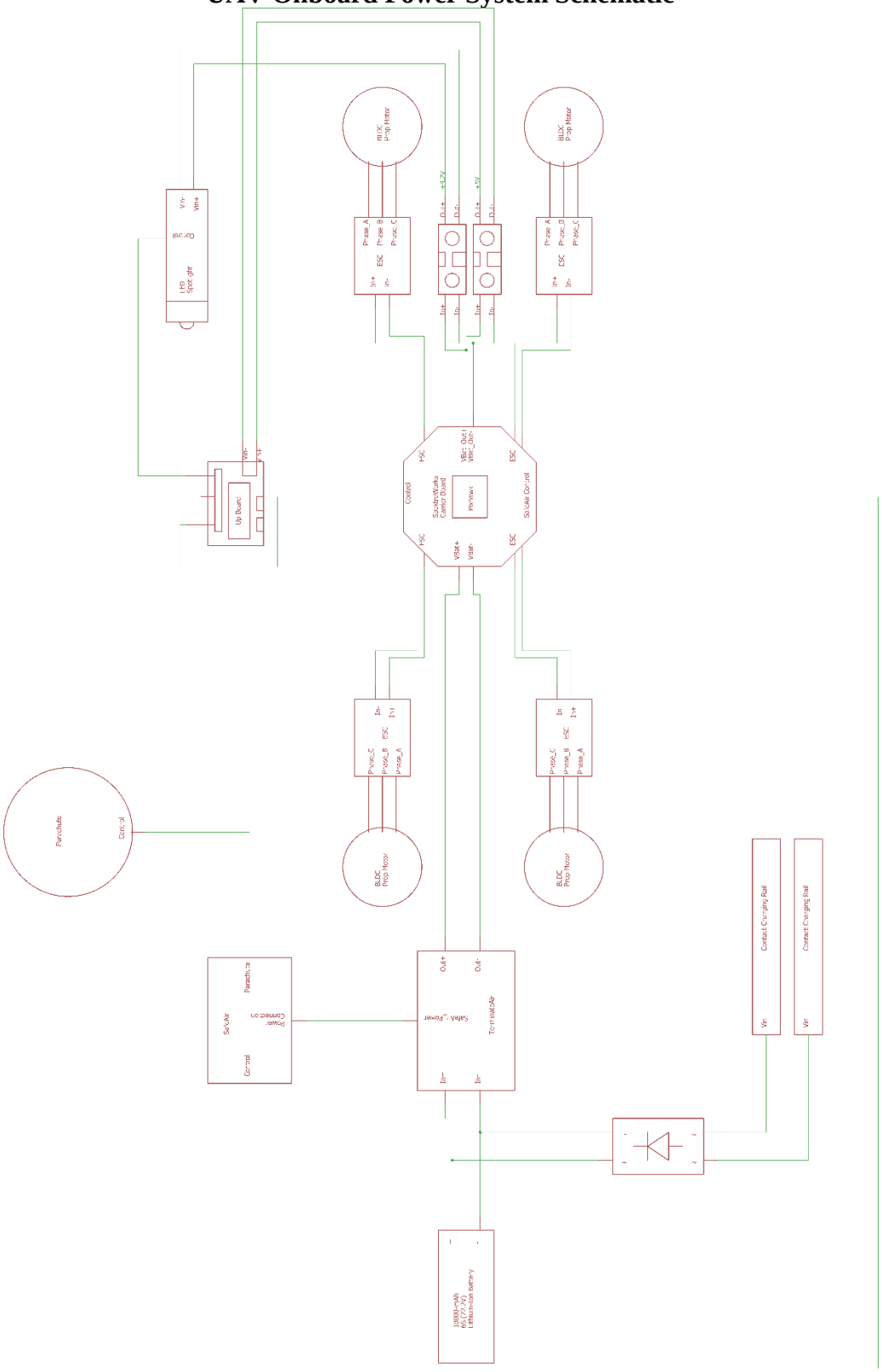
Raspberry Pi Control System



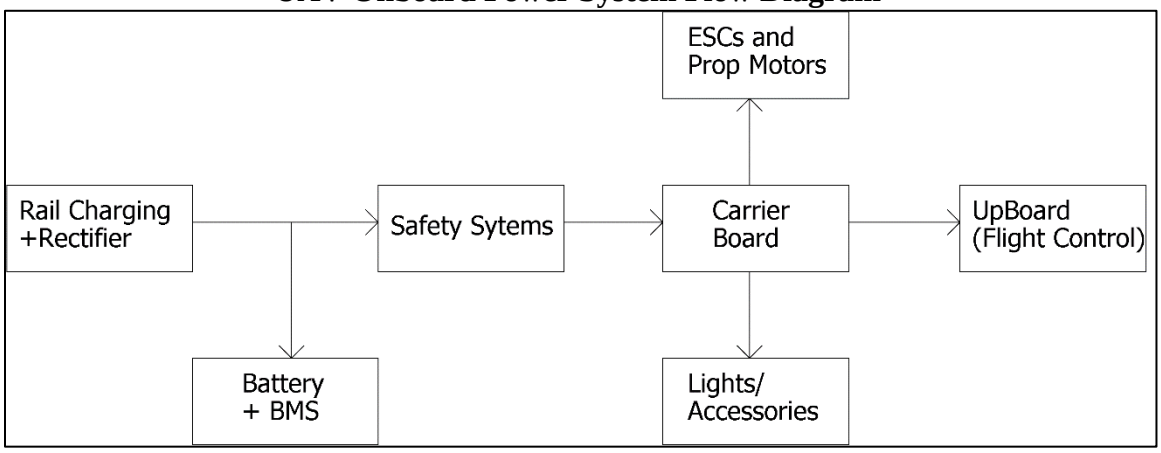
Cooling System



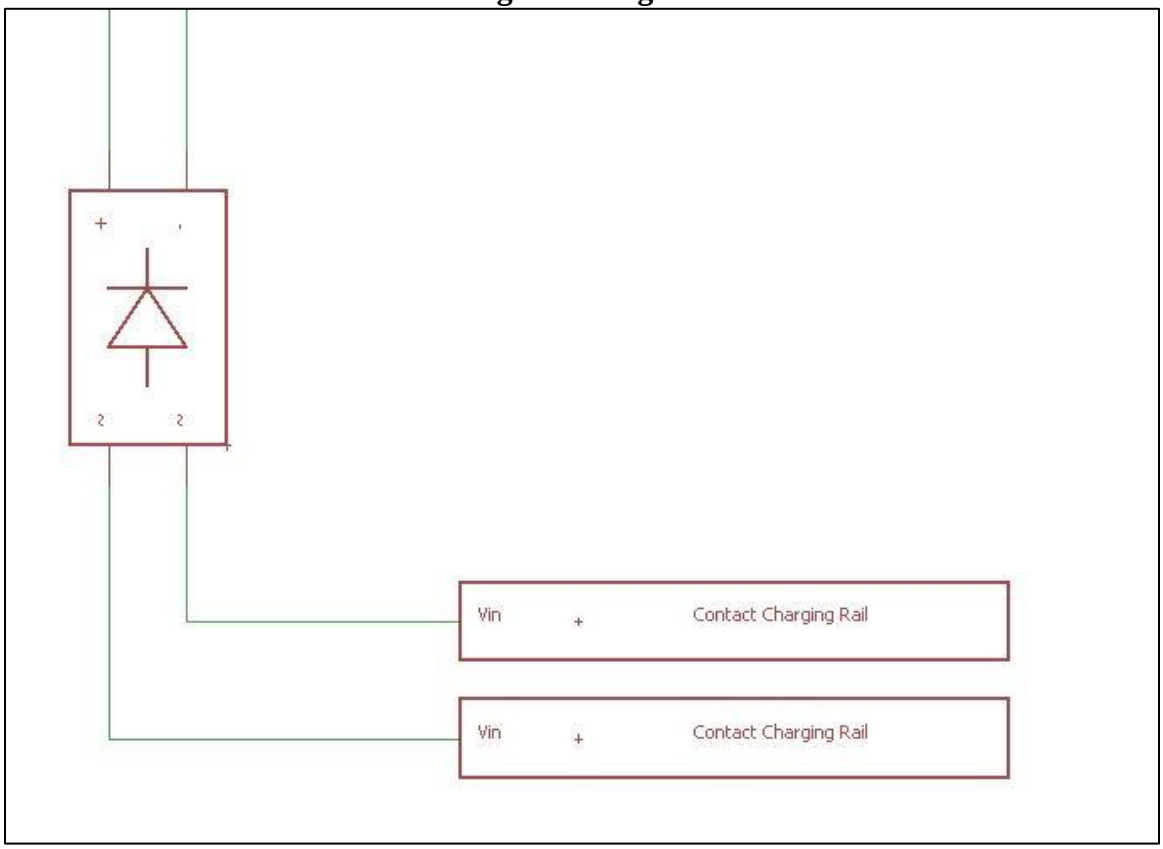
UAV Onboard Power System Schematic



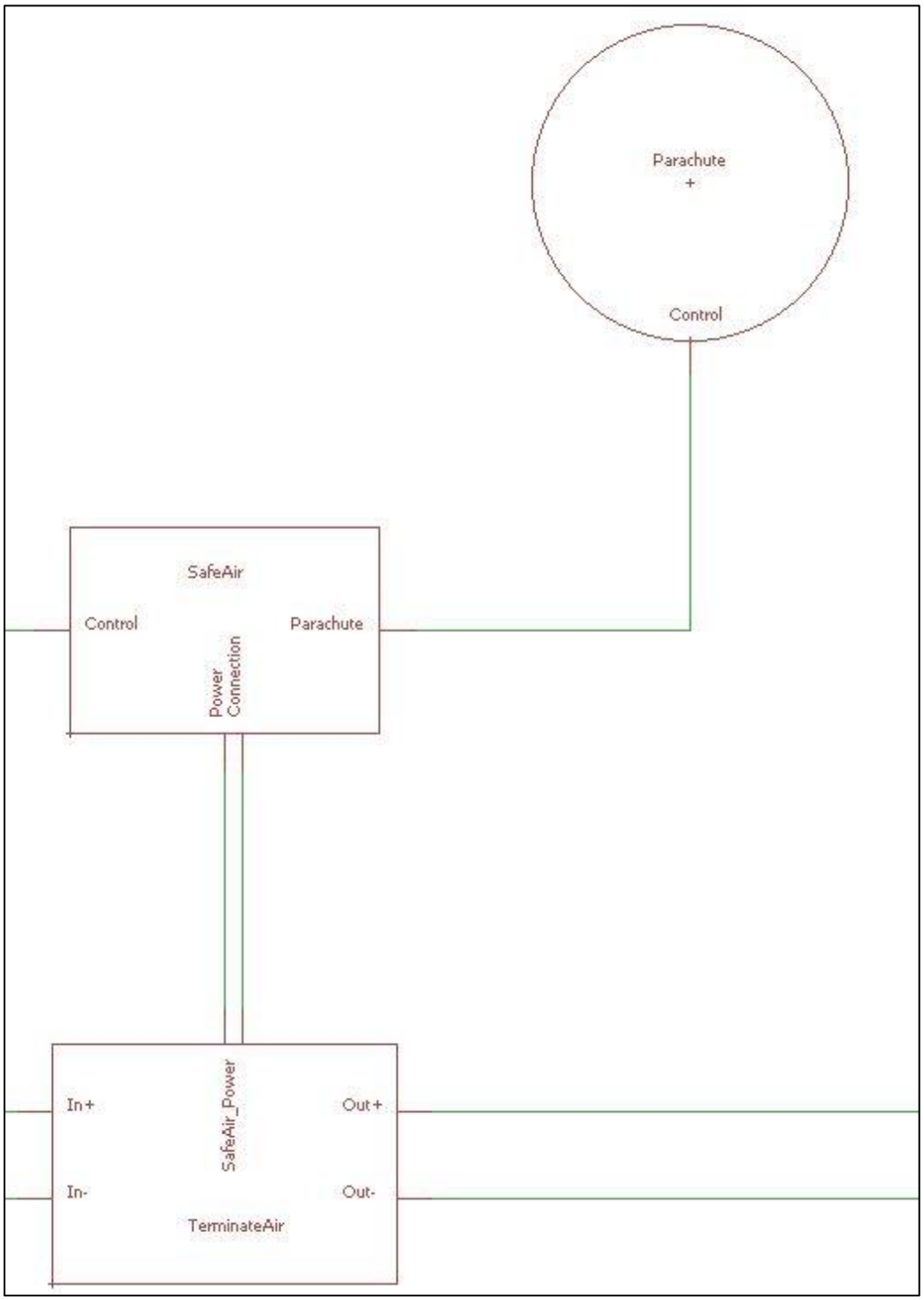
UAV Onboard Power System Flow Diagram



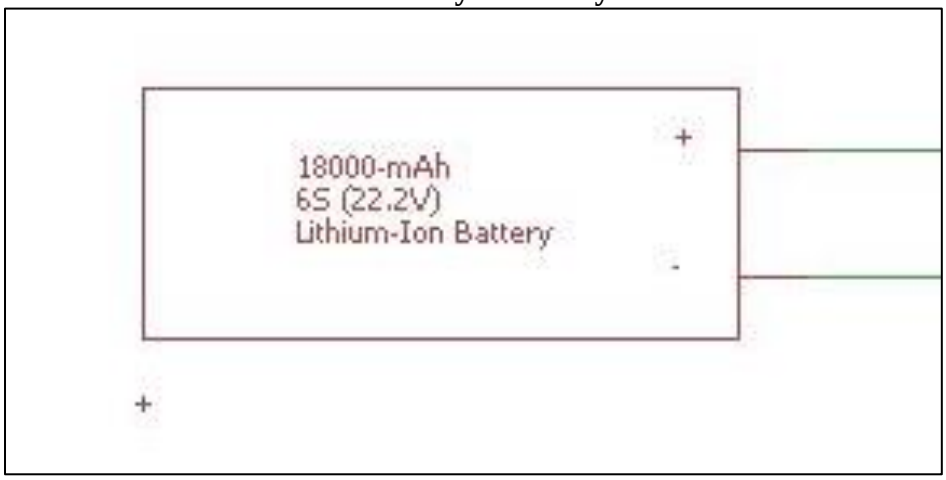
Rail Charger + Bridge Rectifier



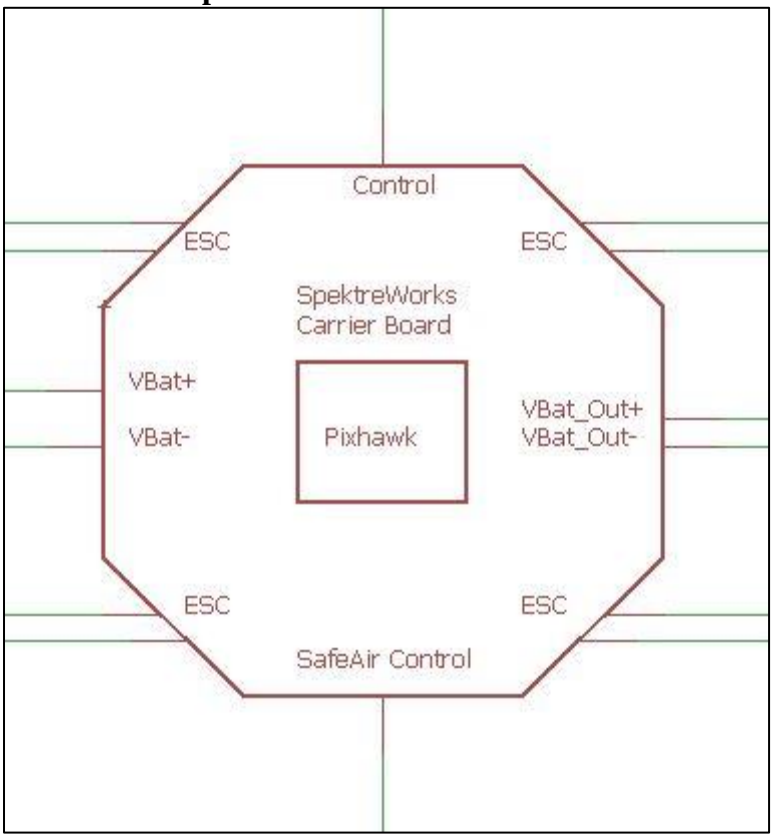
Safety System
SafeAir, TerminateAir, & Parachute



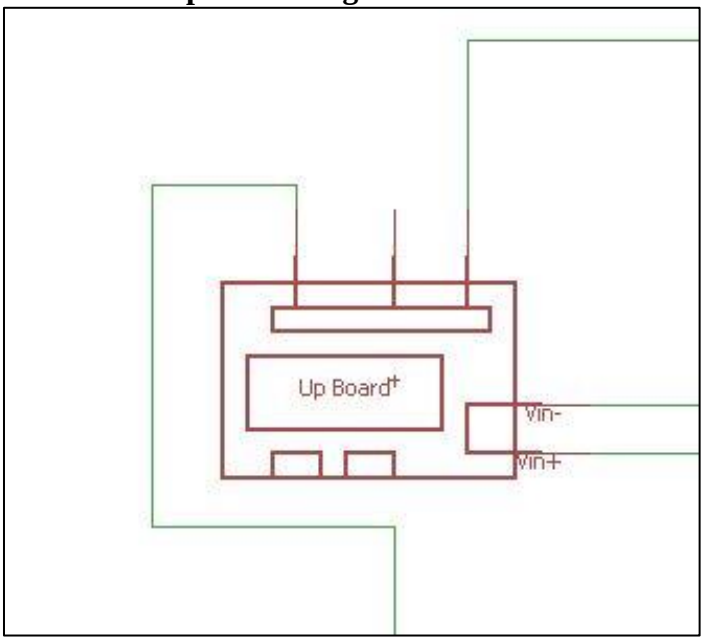
COTS Battery + BMS system



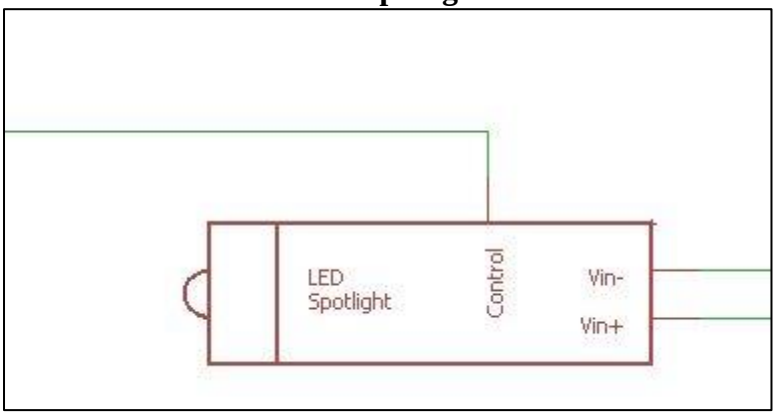
SpektreWorks Carrier Board



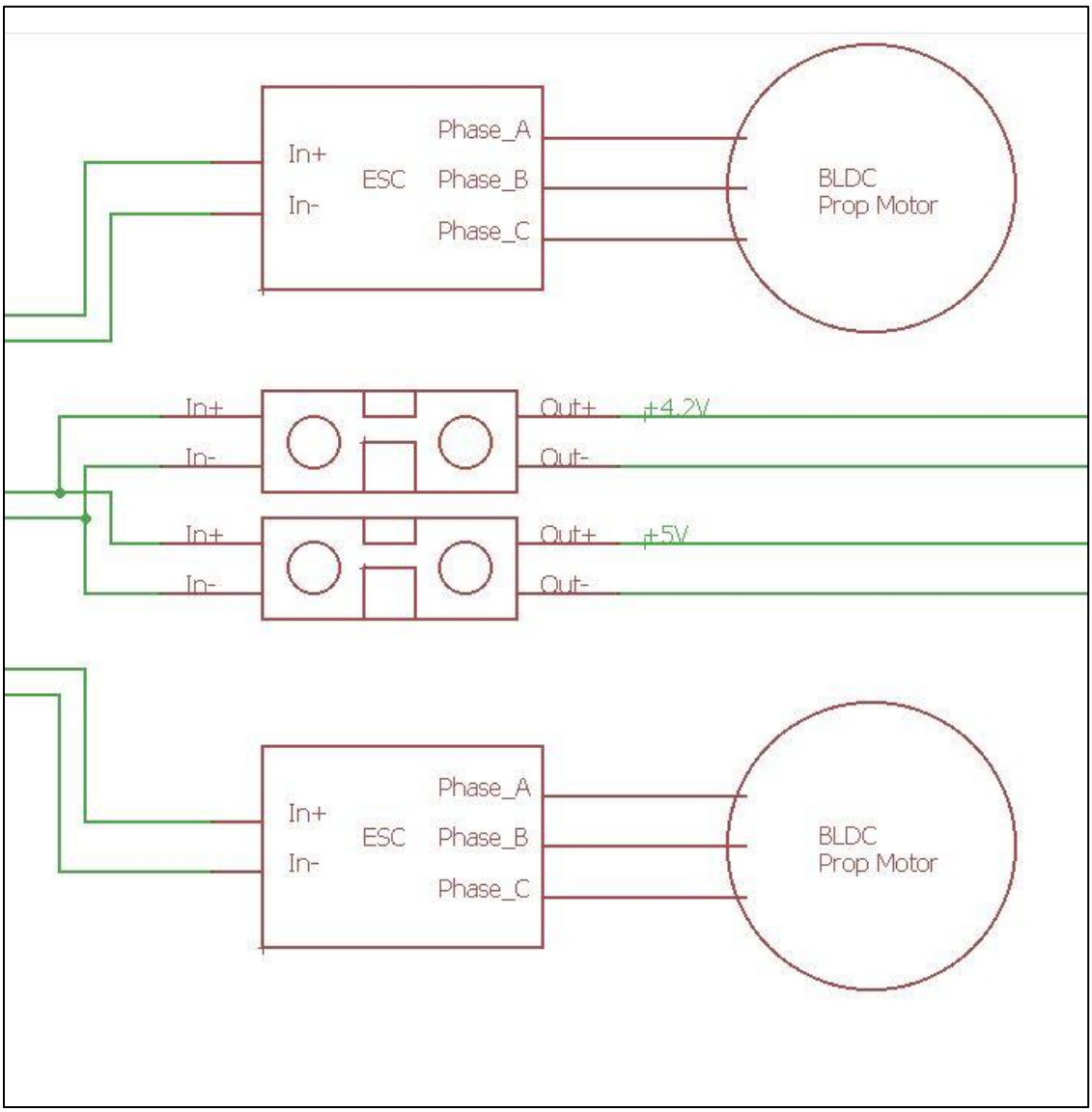
Up Board Flight Controller



LED Spotlight



UAV ESCs and Motors



APPENDIX B

FAA ASU CASE STUDY

Analysis of Government Unmanned Aerial Vehicle Regulations: An ASU Case Study

CEN 590 Independent Study

Sami Mian

Advisor: Dr. Sethuraman Panchanathan

Introduction

As the field of robotics and automation continues to make breakthroughs in advancements and innovative solutions, the use of automated systems is becoming more common place in everyday life. From the introduction of the robot vacuum cleaner, to the integration of automated kiosks at local restaurants, new technologies are being used to help improve the efficiencies of everyday businesses and settings. Currently, one of the most popular topics of interest and innovation is the concept of the "Smart City." Many cities around the world are trying to integrate advanced technologies, such as the Internet of Things (IoT) and automated vehicles, to improve the day-to-day operations of the city ecosystem. Arizona State University (ASU) is unique in that all of Tempe campus can be considered a smart city. Through efforts by the School of Engineering, several research labs are exploring different ways to transform the Arizona State campus to that of a self-contained Smart City. Many of these focuses are based on trying to create innovative solutions to existing problems, or upgrading long-standing technologies to be compliant with today's modern standards.

One of the key attributes that makes a place like Arizona State feel like a community is the sense of safety. ASU is made up of over 100,000 students, teachers, staff members, and other members of the community. One of the concepts we all value is the idea that we feel safe every day on our campus. Because of this, a large emphasis is placed on the security facilities and systems that are currently in use on campus. In addition to having the campus police force, the university is equipped with several different safety services, such as the emergency alarm beacons distributed throughout the major areas and the safety escort service available for students and staff. The emergency beacons, defined

by their iconic blue light, act as two way radios between the boxes and the ASU police station; students can activate them in lieu of dialing 911 for on campus response. The safety escort service is run by the student government, and allows any student or staff member on campus to be given a personal escort after normal business hours across any part of ASU's campuses. Although used extensively, both of these systems are outdated and have various problems. The beacons are connected to the police station through an outdated phone system, and are prone to misuse and accidental calls. Furthermore, average police response time is two to four minutes, which is far too slow to prevent certain types of crime. The safety escort system, which is highly utilized, is often understaffed and cannot meet peak demands during certain times of the year. Sometimes, users have to wait up to 45 minutes for an escort, which dissuades use. Some users also hesitate to use the system even if it's available, because they do not trust the help of strangers, or do not want to be seen using the system. It is the goal of my research to find a way to solve these issues, by introducing automation and innovation to the ASU security theater.

Purpose of Study

One of the main focuses of my research is to work on creating efficient security solutions for a smart city environment. The current vision for a solution is the use of Unmanned Aerial Vehicles (UAVs) for routine surveillance and patrols of a designated area. The use of UAVs is also expected to extend to first responder scenarios as well as safety escorts.

However, in order to autonomously operate UAVs in an urban area, there are regulations and limitations that need to be followed, as well as certifications and proper infrastructure that need to be obtained, all of which are overseen by the Federal Aviation Administration (FAA) [1]. Unlike general aircrafts, which have been supported by long-implemented regulations and procedures, UAVs are a new form of technology, with little oversight and preexisting regulation. The types of classifications, operating procedures, and clearances vary widely based on numerous factors, including location, aircraft size and make, pilot certification, and usage classification [2].

The purpose of this study was to analyze and verify all existing and pending FAA legislation pertaining to UAVs in the study area, and to assess the steps necessary to allow a security system of this sort to operate in an urban setting. Due to the current focus of my research encompassing only the ASU Tempe campus, the main focus of this study is on the necessary FAA regulations for the ASU Tempe campus and surrounding area (downtown Tempe).

Key Terminology

FAA: Federal Aviation Administration

UAV: Unmanned Aerial Vehicles

UAS: Unmanned Aircraft Systems

sUAS: Small Unmanned Aircraft Systems

ATC: Air Traffic Control

Class B Airspace: Airspace from the surface to 9,000 feet MSL (mean sea level) surrounding the nation's busiest airports in terms of IFR operations or passenger enplanements. The configuration of each Class B airspace area is individually tailored and consists of a surface area and two or more layers

Class G Airspace: Uncontrolled airspace, close to the ground (generally under 700 feet)

Current Regulations and Exemption Programs

Part 107

In 2016, the FAA created a new set of rules for unmanned aircraft operations. Formally known as Part 107 of the Federal Aviation Regulations (14 CFR part 107), this set of rules covers a broad spectrum of commercial and hobbyist uses for drones [3]. This set only applies to drones that weight under 55 pounds, and are officially registered with the FAA.

Part 107 rules maintain that all UAS operators must obtain a remote pilot airman certificate, specifically with a UAS rating, or be under the direct supervision of qualified personnel with the certification. In order to obtain this certification, a pilot must:

- Be 16 years old or older
- Pass an initial aeronautical knowledge test at an FAA-approved knowledge center
- Must past a background check verification by the Transportation Safety Administration (TSA) [4]

These are the standard operating rules for a Part 107 license [5]:

- Class G airspace
- Must keep the aircraft in sight (visual line-of-sight
- Must fly under 400 feet
- Must fly during the day
- Must fly at or below 100 mph
- Must yield right of way to manned aircraft

- Must NOT fly over people
- Must NOT fly from a moving vehicle

Operations in class G airspace are allowed unrestricted (under 400 feet). Flying in class B, C, D, and E airspace requires ATC approval [2]. Pilots can operate their aircrafts from 30 minutes prior to sunrise until 30 minutes past sunset. Unmanned aircrafts should always avoid manned aircrafts, and must always be in view of the pilot or a designated visual observer. The aircraft must incorporate proper anti-collision lighting, and minimum weather visibility must be at least three miles from the control station. The maximum speed allowed is 100 miles per hour (87 knots) [5]. Vehicles can carry an external load if it is securely attached and does not adversely affect the flight characteristics or controllability of the aircraft. All of the above requirements can be waiver through a special waiver request for the FAA. A waiver is granted if the pilot(s) can prove that the proposed operation can be conducted safely under the waiver [6].

Section 333

According to federal law, any aircraft operation in the national airspace requires a certificated and registered aircraft, a licensed pilot, and operational approval [7]. Section 333 of the *FAA Modernization and Reform Act of 2012* (FMRA) grants the Secretary of Transportation the authority to determine whether an airworthiness certificate is required for a UAS to operate safely in the National Airspace System (NAS) [7]. This authority is being leveraged to grant case-by-case authorization for certain unmanned aircraft to perform commercial operations prior to the finalization of the Small UAS Rule, which will be the primary method for authorizing small UAS operations once it is complete [7]. This

exception will allow for UAS systems to operate without other FAA requirements in designated areas for an extended amount of time.

The Section 333 Exemption process provides operators who wish to pursue safe and legal entry into the NAS a competitive advantage in the UAS marketplace. In addition to helping companies in this area, the exemption helps discourage illegal operations of UAS systems, and improves overall safety. It is anticipated that the widespread usage of this program will result in significant economic benefits. Due to these benefits, the FAA Administrator has identified this as a high priority project to address demand for civil operations of unmanned systems, for commercial and research purposes [7].

UAS Pilot Program

The UAS Integration Pilot Program is an opportunity for state, local, and tribal governments to partner with private sector entities, such as unmanned systems operators or manufacturers, to accelerate safe UAS integration into society. Entities that have expressed interest in being a Lead Applicant are forming teams and preparing proposals to the FAA to fly more advanced UAS operations, such as beyond visual line-of-sight or night time operations [8].

This program is expected to provide immediate opportunities for new and expanded commercial UAS operations, foster a meaningful dialogue on the balance between local and national interests related to UAS integration, and provide actionable information to the Department of Transportation (DOT) on expanded and universal integration of UAS into the National Airspace System (NAS) [9].

There are two ways to take part in the program. As a Lead Applicant and/or an Interested Party. Lead Applicants must be state, local, or tribal government entities. They will serve as the primary point of contact with the FAA from start to finish. Public Universities will also be able to apply as Lead Applicants in later proposal rounds. Interested Parties are prospective public and private sector applicants/partners OR eligible Lead Applicants. They may submit a request to be on the Interested Parties List to facilitate the formation of Pilot Program teams. Interested parties can be private sector companies or organizations, UAS operators, other stakeholders, or state/local/tribal government entities, including both those that are designated Lead Applicants and those that are not [9].

- The deadline of 2:00 pm ET, November 28, 2017 to submit a Notice of Intent to become a Lead Applicant has passed.
- By 2:00 pm ET, December 13, 2017: Lead Applicants complete volumes I and II in the application portal
- By 2:00 pm ET, December 13, 2017: All interested entities can request to be included on the Interested Parties List
- By 2:00 pm ET, January 4, 2018: Lead Applicants complete volumes III, IV, V, and VI in the application portal

ASU Case Study

In addition to covering the FAA rules and regulations in place for unmanned systems, this report is also focused on analyzing the ability to utilize UAV technologies on the ASU Tempe campus, and addressing any shortcomings that may affect these technologies. Although primarily focusing on the implementation of the Safety Escort & Emergence Response System, this report also looks at general UAV-related research efforts and non-faculty use.

ASU Geography & Environment

The area of focus for this study is the Arizona State University-Tempe campus. Located by downtown Tempe, this campus spans 661 acres, and the main portion stretches from Rio Salado Boulevard (Tempe Town Lake) down to Apache Boulevard, and from Mill Avenue to just past Rural

Road. The area between these four major roadways will encompass the main operations area for the safety escort drones, as well as any research testing, so this is the primary focus of our analysis.

Groups at ASU Interested in UAVs

There are numerous groups with vested interests in receiving approval to operate small aircrafts in the ASU vicinity. These include (but are not limited to):

- ASU Office of Knowledge Enterprise & Development
- ASU Research Enterprise
- ASU Luminosity Lab
- ASU Police Department
- Ira A. Fulton Schools of Engineering
- W.P. Carey School of Business
- Various research groups throughout campus
- Several student organizations involved with robotics and/or drones

Requirements for Safety Escort System

As it stands now, small aircrafts (drones, RC planes, etc.) cannot operate in the airspace over the Arizona State University—Tempe campus. In order to implement the safety escort system at, there are a several regulatory requirements that need to be met or waived.

First and foremost, ASU is in Class B airspace; this is due to the proximity of Phoenix Sky Harbor International Airport, one of the largest airports in the United States. The path of descent for incoming planes is over Tempe Town Lake, which is just North of

the campus. Currently, In order for the university to fly drones in this airspace, the ATC must be notified 24 hours in advance with a flight plan, and the flight must not interfere with airport operations [3]. In addition to the airspace restrictions, the implementation of the autonomous safety escorts violates these Part 107 rules: flying above people, flying at night, flying an unmanned aircraft, flying multiple aircraft, and flying without a visual observer [5]. In order for a system like this to work at ASU, the following Part 107 waivers must be granted to the university:

- 107.29: Daylight Operations (Operating at night time)
- 107.31: Visual line of sight aircraft operations
- 107.33: Visual Observer (separate from pilot)
- 107.35: Operation of multiple small unmanned aircraft systems
- 107.39: Operation over people
- 107.41: Operation over certain airspaces (near airport)

For some of these waivers, proving that operations will be safe is easy. For example, night time flight will be an easy waiver to obtain because of the drone's on-board lighting, as well as the well-lit campus environment in which it will be operating. Other waivers will be more difficult. Specifically, for the restrictions on visual line of sight operations and visual observers, operation over people, and operation of multiple unmanned aircraft systems. We have proposed several technical solutions that can be used to help obtain these waivers.

Technical Solutions for Key Waivers

Visual Observation of Aircraft: Airspace Monitoring Solution

The FAA requires a “visual line of sight” be maintained during drone operation according to Part 107 - Small Unmanned Aircraft Section 107.31. In the regulations, it is stated that the “remote pilot” (drone pilot) or a “visual observer” (designated person to visually observe the drone) must be able to see the aircraft during flight to know the aircraft's location, observe the airspace for other air traffic, and determine that the aircraft will not endanger anybody on the ground. In section 107.33, it states that the “visual line of sight” requirement may be fulfilled by a “visual observer”. Due to the size of the area being covered, it would be more feasible to use visual observers versus having the drone pilot maintain line of sight at all times. To meet the requirements, a “visual observer” is required to be in line of sight of the aircraft, be in communication with the remote pilot at all times, and scan the airspace for “potential collision hazards”. To meet the visual line of sight requirement, three options were explored:

1) Student workers stationed at different places on campus to act as “visual observers”
2) Camera network across campus that could be monitored from a central location where a “visual observer” would be monitoring the cameras
3) Radar system deployed on campus to monitor drones and surrounding airspace

It was found that the option 2 would be the most feasible solution to deploy and the most likely to be approved by the FAA. After a preliminary product search it was found that a number of companies make monitoring systems specifically for drone tracking

applications. As a result, a commercial off the shelf product that can identify and track drones would be easy to deploy. In addition to camera monitoring, many of the systems available also offer a combined camera and radar based tracking system. This is a great option to more accurately determine the location of the drones and to be able to monitor the airspace around them. The system would be monitored at a central command center where the drone operators would be located to meet the requirements specified by section 107.33(a) for the "visual observer" to maintain communication with the "remote pilot". The cost of a tracking system will vary depending on the area being covered but after doing a preliminary cost research, the system could cost between $20,000 and $50,000 for a radar and camera tracking system.

Operation over Unaffiliated Personnel: User Safety

To receive a waiver for autonomous drone flight over other people, we need to demonstrate to the FAA that our system will operate under safe conditions 99% of the time, and has certain safety systems in place as a failsafe. For the safety escort drones planned for use at ASU, we have integrated a gas-propelled parachute system. We are using the ParaZero system, developed by an airspace startup out of Europe [10]. This system includes a SmartAir control system that is able to independent identify loss of control or destabilization of the drone, and deploys an active parachute system to safely recover the drone. The system also comes with an onboard TerminateAir system which instantly cuts the power to all drone propulsion systems, allowing for uninhibited deployment of the parachute. This system has already been tested and approved the FAA as a reliable UAV safety system, and is currently used on several commercial drone platforms.

Operation of Multiple Aircrafts: Unified Command & Control System

In order for the drone system to operate on campus with multiple aircrafts, all of the drones need to be controlled by a central scheduling system. This means that all of the drones need to also be on the same control network. A throughout audit of ASU's existing WiFi system was carried out, and it was determined that the WiFi infrastructure will not be sufficient to host the Command & Control (C2) system for the drone network; it will only be used for low priority, high data transmissions (live video steaming). Instead, a separate network system needs to be setup in order to control and monitor each drone during operation.

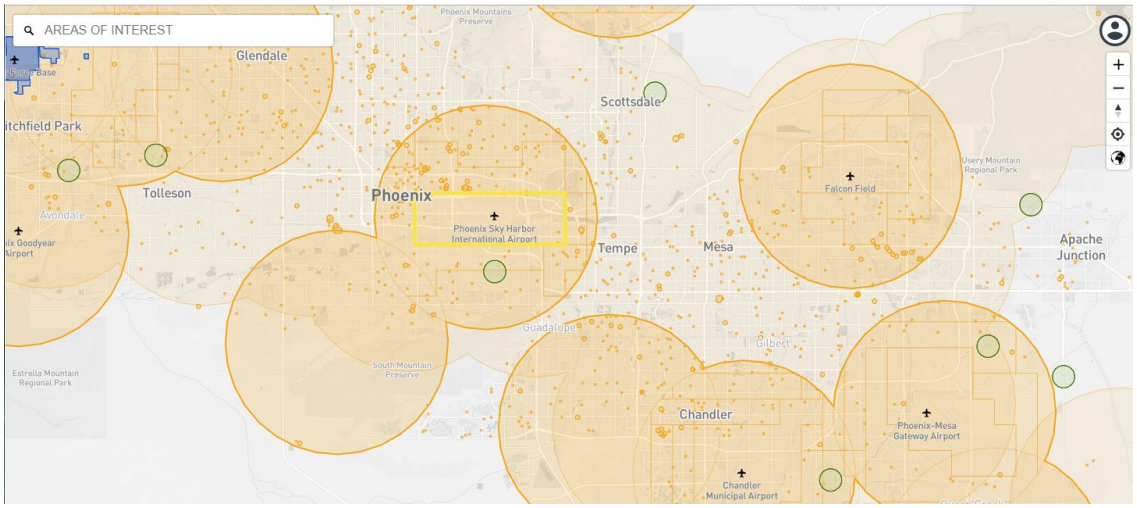
There are two recommended options for the C2 system: piggybacking on existing LTE networks offered by AT&T service provider, or setting up a custom 900 Mhz radio network. For the LTE network, an AT&T sim card and compatible hardware will have to be purchased and integrated onboard each drone. This will cost approximately $200 for the hardware, plus a reoccurring cost of $60 per month, per drone. For the custom network option, it will be implemented using a collection of sector radios placed around campus, all connected to a central server hosted on campus. This system will contain the flight control system that will manage all drone requests and flight plans throughout service. This system will cost about $7,000 in infrastructure.

For the current iteration of the safety escort system, we recommend using an existing data plan through AT&T, at a cost of approximately $60 per drone per month. Once this system scales up the number of agents in use, it will be more cost effective to setup out own network.

Additional Component: Airspace Verification Software

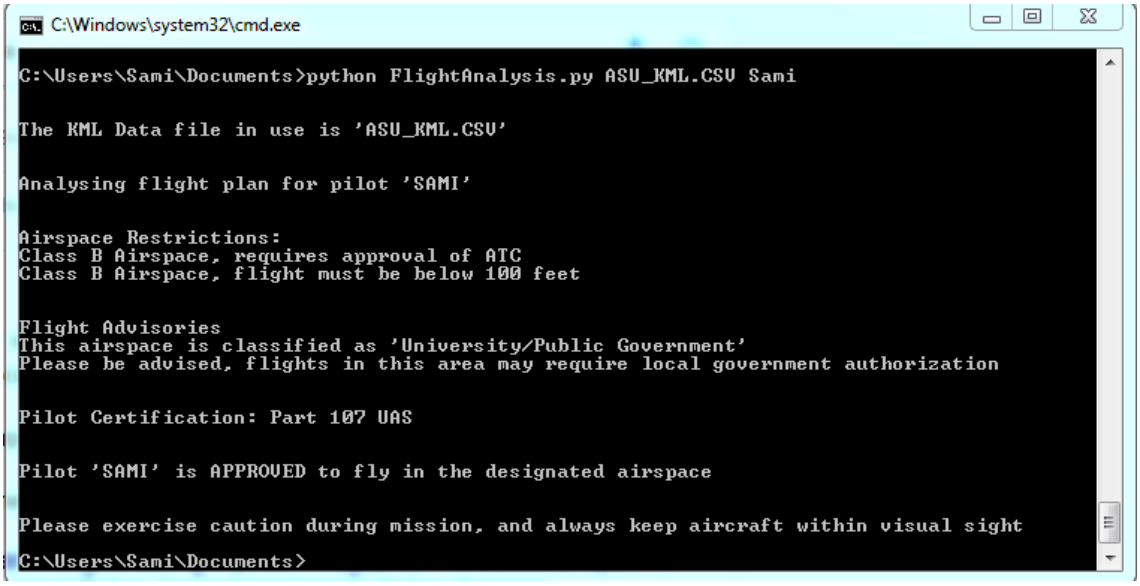
In addition to this in-depth report, I wanted to make a technical contribution towards my thesis work for next semester. In the syllabus for this independent study, I added a possibility for an "A+," by creating a piece of software that would be able to compare drone flight plans against a database of FAA regulations. This software is crucial for any drone fleet management system, because it allows the pilot to verify that their flight plan is legal and will not interfere with any FAA or airport operations. Below are a few screenshots of the application in action. Although the GUI will be discarded, the underlying logic will be integrated into the scheduling system being developed as part of my Master's Thesis.

The first part of this program is a browser-based map interface. This map interface shows the current airspace restrictions in a specified region, and can be used to select a specific area for further analysis. This part of the application was built using the opensource map API created by MapBox [11].



The second part of the program is the flight plan verification tool. This currently runs in the terminal window, as there is no GUI. This piece of the software inputs the flight

plan using KML coordinate data, and isolates the airspace for the flight [12]. It then compares the flight area with all known FAA restrictions and regulations, and returns all known flight restrictions and advisories, based on local and national events and operations. This was originally computed by used a preset database of rules hardcoded into the software, but now the analysis is done using the opensource Airspace API provided my AirMap, LLC [13]. Using this API allows for real time updates to flight advisories and special event restrictions, such as military training exercises or airshows. The KML file with the flight data is stored in the same folder as the program, and can be created from any online map interface (like Google Maps), as well as the browser component of this application. The KML data file can also be stored in another directory; if this is the case, the directory location needs to be added as the first argument in the run command. In the configurations file for this software, the user can detail their pilot's certifications and clearances, as well as several attributes for the drone: weight, flight time, speed, and type. These factors are taken into account when checking the flight plan. The pilot in question can be signified in the second argument in the run command. The first image below shows an example flight plan verification output in the terminal, while the second image is the KML data used for the example. These are the KML coordinates for ASU's airspace.




```
<?xml version="1.0" encoding="UTF-8"?><kml xmlns="http://www.opengis.net/kml/
<Document>
    <Placemark>
        <ExtendedData>
        </ExtendedData>
    <Polygon>
        <outerBoundaryIs>
            <LinearRing>
                <coordinates>-111.93995475769043,33.414714205858296
                -111.92639350891112,33.414750027566235
                -111.92635059356688,33.425424238225716
                -111.93480491638184,33.42556750729568
                -111.93493366241454,33.422021528321615
                -111.94008350372314,33.421985709613296
                -111.93995475769043,33.414714205858296
                </coordinates>
            </LinearRing>
        </outerBoundaryIs>
    </Polygon>
    </Placemark>
</Document>
</kml>
```

Conclusion

Through this independent study, I have had the opportunity to thoroughly study and analyze all the federal and local regulations pertaining to unmanned aerial vehicle operations. As a product of these efforts, we now have a very clear understanding of all the rules and guidance's we will need to follow in order to successfully implement an autonomous drone security system in the ASU Tempe campus area. In addition to having a plan moving forward to fulfilling all regulatory requirements, we also now have several technical solutions in order to address different aspects of unmanned system operations. The technical solutions will allow ASU to expedite the process of acquiring any waivers needed for deployment, and will allow the system to be FAA-compliant and exceed all industry safety standards.

Moving Forward

The next steps for the security drone system is to secure legal permission to deploy the system in Tempe. Originally, the plan was to obtain Part 107 licenses for all operators, as well as all six waivers mentioned earlier. However, as of December 2017, the plan has shifted. Now, through a partnership with ASU Research Enterprise and the Arizona Regional Economic Development Foundation, ASU will be one of the applicants on an Arizona wide proposal to join the FAA's UAS Pilot Program. I will be dedicating the next several months to helping the group craft and submit the joint proposal, and will serve as a subject matter expert on drone operations. In addition to this process, I will also obtain a Part 107 UAS certification in order to allow for interim testing in the surrounding areas. The technical aspects of the drone system will be completed in the Spring: a fully deployable single drone system by the end of January 2018, and a full fleet of autonomous drones by May 2018. By this time, I am confident that ASU will have secured all the necessary legal requirements to make this system a resounding success.

References

[1] https://www.faa.gov/uas/

[2] https://www.faa.gov/uas/media/AC_107-2_AFS-1_Signed.pdf

[3] https://www.faa.gov/uas/media/RIN_2120-AJ60_Clean_Signed.pdf

[4] https://www.faa.gov/uas/getting_started/fly_for_work_business/becoming_a_pilot/

[5] https://www.faa.gov/uas/media/Sec_331_336_UAS.pdf

[6] https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=20516

[7] https://www.faa.gov/uas/beyond_the_basics/section_333/

[8] https://www.faa.gov/uas/programs_partnerships/uas_integration_pilot_program/

[9] https://www.whitehouse.gov/the-press-office/2017/10/25/presidential-memorandum-secretary-transportation

[10] https://parazero.com/product/safeair-m600/

[11] https://www.mapbox.com/api-documentation/

[12] https://github.com/googlemaps/kml-samples

[13] https://developers.airmap.com/v2.0

[14] https://developers.google.com/kml/documentation/kml_tut

APPENDIX C

SOFTWARE CODE EXCERPTS

Python Code for Base Station Operations

```
#
#Below is the main Base Station python script, that runs on bootup
#The purpose of this script is to establish a constant connection
#to the Fleet Management System (FMS) via the cloud API, using Websockets
#This software constantly updates the FMS with internal data, as well as
#pulled from the Arduino Mega, and activates the charging system when giv
#the signal from the FMS.
#
#!/usr/bin/env python3
# External module imports
import RPi.GPIO as GPIO
import time
import numpy as np
import asyncio
import websockets
import pigpio
import smbus


#Place All Setup Code Below

# Pin Definitions:
logicPin = 40 # Pin 40 used for GPIO for Logic Relay Control
I2C_ADDR=9 #Pin for i2C communication to Arduino Mega


# Pin Setup:
GPIO.setmode(GPIO.BOARD) # Broadcom pin-numbering scheme
GPIO.setup(logicPin, GPIO.OUT) # LED pin set as output


# Initial state for Login Pin Out:
GPIO.output(logicPin, GPIO.LOW)

activate = False;  #Boolean variable used to turn on charging system


#Code below is for websockets for web API communication


async def consumer(message):
    print("> {}".format(message))


async def consumer_handler(websocket):
    while True:
        message = await websocket.recv()
        await consumer(message)
```

```
async def producer_handler(websocket):
    while True:
        # message = await producer()
        await websocket.send("hello")
        await asyncio.sleep(5)


async def handler():
    async with websockets.connect('ws://localhost:1323/ws') as websocket:
        consumer_task = asyncio.ensure_future(consumer_handler(websocket))
        producer_task = asyncio.ensure_future(producer_handler(websocket))
        sensor_handler = asyncio.ensure_future(sensor_handler(websocket))
        done, pending = await asyncio.wait(
                [consumer_task, producer_task],
                return_when=asyncio.FIRST_COMPLETED,
                )

        for task in pending:
            task.cancel()



async def sensor_handler():
    while True:
        if(activate = "on"):
            GPIO.output(logicPin, GPIO.HIGH)
            time.sleep(10);

        if(activate = "off"):

            GPIO.output(logicPin, GPIO.LOW)
            time.sleep(10);


async def arduino_handler():
    while True:
        bus = smbus.SMBus(1)
        address = 0x04


        while True:
        request = "openConn"
        if not request:
        continue

        writeData(request)        #Send data to Arduino
        # sleep one second
        time.sleep(1)
```

```
async def i2c_handler(id, tick):
    global pi

    s, b, d = pi.bsc_i2c(I2C_ADDR)

    if b:
        print(d[:-1])



async def writeData(value):
    bus.write_byte(address, value)
    # bus.write_byte_data(address, 0, value)
    return -1

async def readData():
    number = bus.read_byte(address)
    # number = bus.read_byte_data(address, 1)
    return number


asyncio.get_event_loop().run_until_complete(handler())
```

Arduino Code for Base Station Diagnostics

```
//Import the library required
#include <Wire.h>

//Slave Address for the I2C Communication
#define SLAVE_ADDRESS 0x04

// Define the pin to which the temperature sensor is connected.
const int pinTemp = A0;
const int pinLight = A1;

//Moisture Sensor Setup Code
int sensorPin = A0; // select the input pin for the potentiometer
int sensorValue = 0; // variable to store the value coming from the sensor7

// Define the B-value of the thermistor.
// This value is a property of the thermistor used in the Grove - Temperature Sensor,
// and used to convert from the analog value it measures and a temperature value.
const int B = 3975;

// Define the pins to which the sound sensor and LED are connected.
const int pinSound = A2;
const int pinLed   = 7;
const int relayPin =  8;
int thresholdValue = 500;

char number[50];
int state = 0;
int data[6];


//Code Initialization
void setup() {
    // initialize i2c as slave
    Serial.begin(9600);
    Wire.begin(SLAVE_ADDRESS);
    // define callbacks for i2c communication
    Wire.onReceive(receiveData);
    //  Wire.onRequest(sendData);

    pinMode(pinLed, OUTPUT);  //Configure LED pin for status indicator

    // Configure the relay's pin for output signals.
    pinMode(relayPin, OUTPUT);

}
```

```
void loop() {

   // Get the (raw) value of the temperature sensor.
    int val = analogRead(pinTemp);

    // Determine the current resistance of the thermistor based on the sensor value.
    float resistance = (float)(1023-val)*10000/val;

    // Calculate the temperature based on the resistance value.
    float temperature = 1/(log(resistance/10000)/B+1/298.15)-273.15;

    // Read the value of the sound sensor.
    int sensorValue = analogRead(pinSound);

    // Read the value of the light sensor. The light sensor is an analog sensor.
    int sensorValue = analogRead(pinLight);

    //Read the value of the moisture sensor
    int moistureSensorValue = analogRead(sensorPin);

    data = [temperature, sensorValue, moistureSensorValue]
  delay(100);
} // end loop

// callback for received data
void receiveData(int byteCount) {
  int i = 0;
  while (Wire.available()) {
    number[i] = Wire.read();
    i++;
  }
  number[i] = '\0';
  Serial.print(number);
}  // end while

// callback for sending data
void sendData(input) {
  Wire.write(input);
}

//End of the program
```